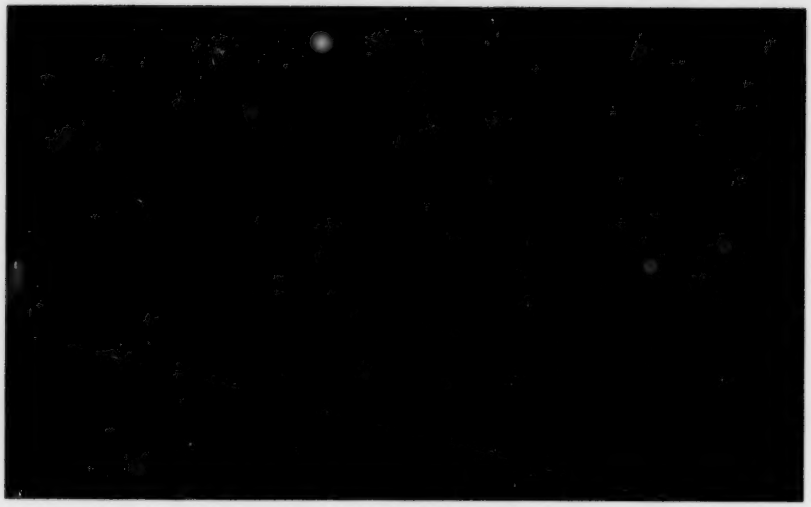
CENTRE DE RECHERCHE
SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE



MÉMOIRE D'ÉTUDIANT

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A,
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: 05-24146 BIBPOLYTEC
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

CRT-681

UNE APPROCHE DE RESTRICTION BASÉE SUR UNE MÉTHODE PRIMALE-DUALE POUR RESOUDRE LE PROBLÈME DE LOCALISATION-DISTRIBUTION AVEC ECHANGES INTERDEPOTS

par

Nicole Tétreault

Mémoire de maîtrise, département d'Informatique et de Recherche opérationnelle, Université de Montréal, dirigé par Teodor Gabriel Crainic.

Centre de recherche sur les transports - Publication #681
Janvier 1990

90-03011/

Dépôt légal - 1er trimestre 1990

SOMMAIRE

Nous présentons dans ce mémoire un algorithme servant à résoudre un problème particulier que pose l'administration d'une flotte de véhicules. Il s'agit de la problématique entourant la localisation des dépôts ainsi que la gestion des mouvements à vide. Comme apport théorique, nous dérivons deux nouvelles formulations mathématiques du modèle qui nous permettent de développer une méthode de restriction à laquelle se greffe une adaptation d'un algorithme de relaxation primal-dual. L'originalité du travail de recherche provient de l'approche adoptée pour la résolution du problème.

Le problème à résoudre se trouve à l'origine d'un important problème auquel font face les grandes compagnies de transport maritime et terrestre de conteneurs. Les compagnies distribuent à leurs clients des conteneurs de divers biens, par différents modes de transport. Les clients déchargent alors les conteneurs et les renvoient aux dépôts. Par ailleurs, les clients commandent aussi des conteneurs vides aux dépôts qu'ils chargent et retournent aux dépôts. Or, certains dépôts reçoivent plus de conteneurs qu'ils n'en envoient et vice versa, engendrant un déséquilibre de conteneurs vides dans le réseau. Pour rétablir l'équilibre, il est alors nécessaire de procéder à des mouvements à vide des conteneurs entre les dépôts. Ce problème connaît d'autres applications, notamment dans le domaine du camionnage, des réseaux ferroviaires, de la location d'équipement, etc.

Un tel problème se formule mathématiquement comme un modèle de localisation-distribution multiniveau, multimode, multiproduit avec contraintes supplémentaires de flot qui assurent le maintien de l'équilibre des conteneurs dans le réseau. L'objectif visé est la minimisation des coûts d'ouverture des dépôts et des coûts de transport résultant des mouvements à vide des conteneurs entre les dépôts ainsi que ceux bidirectionnels entre les dépôts et les clients.

Dans cette étude, nous étudions le modèle déterministe, à coûts linéaires, sans capacités ni sur les dépôts ni sur les modes de transport. De plus, nous considérons seulement les mouvements à vide des conteneurs. Nous supposons aussi qu'il n'existe pas de mouvements interclients et que les mouvements interdépôts sont plus efficaces que les mouvements entre les dépôts et les clients, justifiant ainsi la présence des mouvements entre les dépôts.

La résolution d'un tel problème s'avère une tâche complexe à cause de sa grande taille et de ses contraintes d'intégralité imposées sur certaines variables. Or, nous ne nous attaquons pas directement à la formulation initiale du problème, mais nous étudions plutôt la relaxation forte du problème qui possède une structure de réseau propice aux développements algorithmiques.

La méthodologie que nous proposons est la suivante. Nous extrayons d'abord, de la formulation de la relaxation forte, un problème équivalent dans lequel apparaissent uniquement des variables de flot et qui possède lui aussi une structure de réseau. Nous appliquons alors, au problème transformé, un procédé de restriction qui consiste à imposer des capacités sur certaines variables, restreignant ainsi le domaine réalisable du problème. Nous définissons alors un sous-problème restreint, dans lequel nous retrouvons également une structure de réseau, que nous résolvons par une adaptation d'un algorithme de relaxation primal-dual. Une telle approche génère donc une borne supérieure sur la valeur de la fonction objectif du problème initial. Dans cette étude, nous présentons deux versions algorithmiques. La première, à caractère glouton, change le domaine réalisable restreint dès qu'une descente possible s'annonce dans cette direction. Tandis que la deuxième, à l'opposé, n'effectue qu'une modification locale de la portion du domaine réalisable sur laquelle se fait la

résolution du sous-problème restreint.

Quelques expérimentations numériques ont été menées dans le but de déterminer la validité de notre algorithme, ainsi que la qualité de la solution qu'il génère. Nous avons testé une seule des deux versions sur des problèmes dont nous connaissions la solution optimale. Les résultats obtenus jusqu'ici se révèlent encourageants quoique non-concluants. En effet, la solution générée par notre algorithme, lorsque non-optimale, se situe en-deça de 2,4% de la solution optimale. Cependant, dans certains cas, l'algorithme ne semble pas converger. Or, nous croyons que la seconde version, présentement en voie de développement, pourra remédier à cette difficulté.

Ainsi, nous poursuivrons nos travaux de recherche qui consistent essentiellement à compléter et améliorer les versions actuelles. Nous comptons alors pouvoir effectuer de plus amples tests sur des problèmes de plus grande taille, en particulier, sur des données réelles provenant d'une compagnie européenne de transport de conteneurs.

Mots-clés : localisation-distribution, contraintes d'équilibrage, mouvements à vide, Bertsekas, méthode restrictive.

TABLE DES MATIERES

SOMMAIRE iii

LISTE DES TABLEAUX viii

LISTE DES FIGURES ix

INTRODUCTION 1

CHAPITRE I MODELE

I.1 Notations et définitions 6
I.2 Formulation mathématique 8
I.3 Relaxations 11
I.4 Stratégies de résolution 13

CHAPITRE II METHODES DE RELAXATION PRIMALES-DUALES

II.1 Coûts linéaires 18
II.1.1 Problème 18
II.1.2 Problème dual et conditions d'optimalité 19
II.2.3 Notations et définitions 22
II.1.4 Algorithme 27

II.2 Coûts convexes 29
II.2.1 Problème 30
II.2.2 Problème dual 31
II.2.3 Notations et définitions 33
II.2.4 Algorithme 36
II.2.5 Convergence de l'algorithme 38

CHAPITRE III APPROCHE DE RESTRICTION

III.1 Approches de résolution 43
III.2 Problème transformé 44
III.3 Sous-problème restreint et propriétés 48

III.4 Définitions et notations 52
III.5 Problème dual du sous-problème restreint 54
III.6 Autres définitions et notations 57
III.7 Algorithme 60
III.8 Particularités de l'algorithme 64
III.8.1 Lors du marquage des noeuds 66
III.8.2 Sur la chaîne d'augmentation de flot 68
III.8.3 Pas de descente lors d'un pas dual 73
III.8.4 Modifications du domaine réalisable restreint 75
III.8.5 Ajustement des coefficients d'ouverture des dépôts 75
III.8.5.1 Dans la version globale 76
III.8.5.2 Dans la version locale 79
III.8.6 Changement des variables restrictives 79
III.9 Conclusion 82

CHAPITRE IV EXPERIMENTATIONS NUMERIQUES

IV.1 Problèmes-test 84
IV.2 Programme 85
IV.2.1 Version globale 86
IV.2.2 Implantation de l'algorithme 89
IV.3 Résultats et commentaires 91

CONCLUSION 98

BIBLIOGRAPHIE x

APPENDICES xiii

REMERCIEMENTS xxxiii

LISTE DES TABLEAUX

Tableau I	Bornes-ε pour les différents types d'arcs connectés au dépôt *j* pour un produit *p* donné	59
Tableau II	Bornes-ε modifiées	72
Tableau III	Ajustements des bornes-ε suite à une modification des prix	74
Tableau IV	Quatre cas nécessitant l'ajustement du coefficient d'ouverture d'un dépôt *j* après une modification du flot dans la version globale	78
Tableau V	Trois critères de sélection de la variable restrictive relative à un dépôt *j* donné	81

LISTE DES FIGURES

Figure 1	**Structure de réseau présente dans le modèle**	9
Figure 2.1	**Graphe de la fonction linéaire de coût primale associée à un arc (i, j)**	21
Figure 2.2	**Graphe de la fonction linéaire de coût duale associée à un arc (i, j)**	22
Figure 2.3	**Graphe de la fonction convexe de coût primale associée à un arc (i, j)**	32
Figure 2.4	**Graphe de la fonction convexe de coût duale associée à un arc (i, j)**	32
Figure 3.1	Graphe de la fonction de coût associée à un arc interdépôt pour un produit p donné	51
Figure 3.2	Graphe de la fonction de coût associée à un arc restreint d'un dépôt j pour un produit p donné	52
Figure 3.3	Graphe de la fonction de coût associée à un arc restrictif d'un dépôt j	52
Figure 3.4	Graphe de la fonction duale de coût associée à un arc interdépôt pour le produit p	55
Figure 3.5	Graphe de la fonction duale de coût associée à un arc restreint d'un dépôt j pour le produit p	56
Figure 3.6	Graphe de la fonction duale de coût associée à un arc restrictif d'un dépôt j	56
Figure 3.7	Organigramme de l'algorithme de restriction	63
Figure 3.8	Trois cas entraînant la violation de la définition de la variable restrictive d'un dépôt j suite à une modification de flot	70

INTRODUCTION

Nos recherches qui se trouvent à l'origine d'un important problème auquel font face les grandes compagnies de transport maritime et terrestre de conteneurs. Ce problème représente un cas particulier que pose l'administration d'une flotte de véhicules, soit celui de la localisation des dépôts et de la gestion des mouvements à vide des conteneurs. Il s'agit du problème de localisation-distribution avec échanges interdépôts. Crainic, Dejax et Delorme (1986) ont fait une analyse approfondie de ce problème et l'exposent ainsi :

> La compagnie de transport de conteneurs livre à ses clients des conteneurs pleins de divers types, par des modes (rail, route, mode combiné, et autres.) dont les caractéristiques de coût, de temps, etc. sont connues. Pour chaque client la compagnie connaît le nombre de conteneurs pleins de chaque type à livrer. Lorsque reçus, les clients déchargent les conteneurs, qui sont retournés à un dépôt. Les clients commandent aussi, pour leurs expéditions, des conteneurs vides aux dépôts. Ces demandes sont également connues par la compagnie. Une fois chargés, les conteneurs sont dirigés vers leur destination, normalement un autre point du réseau de la compagnie, directement ou via leur dépôt d'origine ou un autre. Le problème est compliqué du fait que les demandes des clients entraînent un déséquilibre dans les disponibilités de conteneurs vides entre les divers points du réseau de la compagnie : certaines zones manquent de certains types de conteneurs, tandis que des surplus sont enregistrés ailleurs. Cette situation impose des transferts de conteneurs vides entre les dépôts afin de rééquilibrer le réseau. Une telle opération génère des coûts de transport importants dont il faut tenir compte lors de la résolution du problème de localisation. En effet, il s'agit de localiser les dépôts des conteneurs vides pour satisfaire les demandes et minimiser les coûts totaux de structure, de fonctionnement et de transport.

Ce problème constitue un problème de planification stratégique du réseau logistique de distribution de la compagnie. Il s'agit de localiser les dépôts de conteneurs et optimiser les mouvements à vide des conteneurs qui occupent une place importante dans les considérations logistiques, en particulier de localisation des dépôts (Dejax et Crainic, 1986). En effet, de nos jours, à cause de la perpétuelle inflation des coûts de transport et aux mouvements à vide n'engendrant aucun revenu, ces derniers, lorsque diminués, représentent une économie certaine pour la compagnie. La résolution d'un tel problème par des techniques de recherche opérationnelle permettra donc à la compagnie de transport de mieux gérer les mouvements à vide de ses conteneurs et ainsi réduire les coûts non-négligeables qu'ils entraînent. Le but poursuivi par ce travail de recherche est de présenter un algorithme de résolution pour ce problème spécifique de localisation afin d'obtenir une solution optimale, ou sinon une borne sur cette solution qui pourrait être utilisée dans un algorithme d'énumération.

Dans le cadre de cette étude, nous ne considérons que les mouvements à vide. De plus, nous faisons l'hypothèse qu'il s'agit d'un modèle déterministe (offres et demandes connues), à coûts linéaires et sans capacités ni sur les dépôts ni sur les modes de transport. Nous supposons également que les conteneurs vides transitent obligatoirement par un dépôt et qu'il n'y a aucun mouvement interclient. Aussi, nous supposons que les mouvements interdépôts sont plus efficaces que les mouvements entre les dépôts et les clients, justifiant ainsi les mouvements entre les dépôts.

Les éléments de modélisation ainsi que leurs propriétés ont été étudiés en détail par Crainic, Dejax et Delorme (1986, 1989). Le modèle, dans lequel sont définis deux ensembles de variables de décision : binaires pour ouvrir les dépôts, et continues pour le flot, se formule mathématiquement comme un problème de localisation de dépôts multiniveau, multimode, multiproduit avec échanges interdépôts.

Un tel problème s'avère difficile à résoudre à cause de sa grande taille et de ses contraintes d'intégralité imposées sur certaines variables. D'autre part, la formulation telle que présentée exhibe, par la présence des contraintes supplémentaires de flot servant à rééquilibrer le réseau, une structure sous-jacente de réseau propice aux développements algorithmiques. Outre l'objet de cette étude, deux autres méthodes de

résolution basées sur différentes variantes du modèle ont été développées et programmées, notamment un algorithme d'énumération de type "branch-and-bound" (Hodgson, 1988) et un algorithme de type "flip-flop" basé sur des procédures duales (Delorme, Crainic et Dejax, 1988). Le premier génère une borne inférieure sur la valeur de l'objectif du problème initial, tandis que le deuxième produit à la fois une borne inférieure, de meilleure qualité, ainsi qu'une borne supérieure inintéressante, car son comportement est imprévisible.

L'approche de résolution que nous présentons dans cette étude vise à exploiter la structure de réseau inhérente à la relaxation forte du problème. Dans cette perspective, nous proposons de résoudre cette dernière en développant un algorithme basé sur une méthode de relaxation primale-duale (Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986). Les étapes qui ont mené à l'approche développée sont les suivantes. Nous extrayons d'abord une formulation équivalente à celle de la relaxation forte dans laquelle apparaissent seulement des variables de flot. Au problème transformé, qui possède lui aussi une structure de réseau, nous appliquons un procédé de restriction. Celui-ci consiste à imposer des contraintes de capacités sur certains arcs, restreignant ainsi le domaine réalisable du problème à solutionner. Nous définissons alors un sous-problème restreint dans lequel nous retrouvons de nouveau une structure de réseau que nous résolvons par une adaptation de l'algorithme de relaxation primal-dual. Nous présentons dans cette étude deux versions algorithmiques. La première, à caractère glouton, consiste à changer de façon drastique le domaine réalisable restreint dès qu'une descente dans cette direction s'annonce possible. Tandis que l'autre, à l'opposé, ne permet qu'une modification locale de la portion du domaine réalisable sur laquelle se fait la résolution du sous-problème restreint. Ce dernier génère une borne supérieure sur la valeur optimale du problème originel étant donné l'ajout des contraintes de capacités. Cette borne pourra être utilisée conjointement à la borne inférieure obtenue par l'algorithme "flip-flop" dans l'algorithme du "branch-and-bound", lui assurant ainsi une meilleure convergence.

Ce mémoire se présente de la façon suivante. Tout d'abord, le chapitre un sert à rappeler les principaux éléments de modélisation du problème que nous utiliserons

dans ce travail. Nous y discutons également des différentes stratégies de résolution élaborées. Dans le chapitre deux, nous décrivons les méthodes de relaxation primales-duales pour les coûts linéaires et convexes (Bertsekas, 1985; Bertsekas et Tseng, 1985, 1988; Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986; Tseng et Bertsekas, 1987). Celle pour les coûts convexes sera adaptée à notre problème bien particulier pour résoudre les différents problèmes générés par l'approche de restriction. Le chapitre trois constitue le noyau de ce travail. Nous y présentons la méthode que nous proposons pour solutionner le problème de localisation-distribution avec échanges interdépôts. Outre une description du cheminement suivi pour développer l'approche de restriction, nous détaillons les différentes étapes des deux versions algorithmiques. Finalement, dans le dernier chapitre, nous discutons des expérimentations numériques effectuées pour tester une des deux versions de l'algorithme que nous avons programmée. Finalement, nous concluons en faisant une synthèse regroupant les objectifs de ce travail ainsi que les résultats obtenus, puis nous émettons des avenues futures de recherche.

Par abus de langage, tout au long de ce travail, du fait que les contraintes de conservation de flot modélisent le maintien de l'équilibre dans le réseau, nous référons fréquemment à ces contraintes en utilisant la terminologie ''contraintes d'équilibrage''.

CHAPITRE I

MODELE

Dans ce premier chapitre, nous rappelons certains éléments de modélisation extraits de Crainic, Dejax et Delorme (1986, 1989), concernant le problème de localisation-distribution avec échanges interdépôts que nous nous proposons d'étudier.

Tout d'abord, nous introduisons les notations et définitions. Puis, nous présentons le modèle linéaire non-restrictif dans lequel tout client peut être en relation avec autant de dépôts qu'il le désire, ainsi que deux variantes qui nous seront utiles au chapitre III. Finalement, nous discutons des stratégies de résolution déjà développées et de celles envisagées.

Dans le cadre de ce mémoire, nous ne considérons que les mouvements à vide. Ces derniers, générant des coûts de transport considérables et n'engendrant aucun revenu, occupent une place importante dans le problème de la localisation des dépôts (Dejax et Crainic, 1986). Ainsi, réduire les mouvements à vide assure une économie certaine pour la compagnie de transport de conteneurs. Aussi, nous supposons que les données (offres et demandes) sont connues. Selon cette hypothèse, il s'agit d'un modèle déterministe. Nous supposons également que les conteneurs vides transitent obligatoirement par un dépôt et qu'il n'y a aucun mouvement interclient. Finalement, nous n'étudions que le modèle à coûts linéaires non-négatifs (englobant les coûts de transport ainsi que ceux d'ouverture et d'opération des dépôts) et sans capacités ni

sur les dépôts ni sur les modes de transport.

Outre une discussion plus approfondie des hypothèses, nous retrouvons une étude exhaustive du modèle, incluant différentes variantes ainsi que leurs propriétés dans Crainic, Dejax et Delorme (1986, 1989).

I.1 Notations et définitions

Puisque notre problème est linéaire et qu'on ne suppose pas de capacités explicites de transport, le trafic utilise le moyen de transport le moins coûteux. Ainsi, pour simplifier la notation, nous n'utilisons dans ce qui suit qu'un seul arc dans chaque direction, soit celui représentant le mode ou la combinaison des modes de transport de plus bas coût unitaire. Nous négligeons donc l'indice représentant le mode dans la notation des paramètres du modèle.

Soit

$$C = \{\, i \mid \text{client} \,\}$$

$$D = \{\, j \mid \text{dépôt candidat} \,\}$$

$$P = \{\, p \mid \text{type de conteneur vide} \,\}.$$

Ces éléments forment un réseau conceptuel $G = (N, A)$, souvent parallèle à un réseau physique. Les noeuds, ou sommets, sont les clients et les dépôts candidats : $N = C \cup D$. Les liens, ou arcs, représentent les mouvements possibles de chaque type de conteneur entre les dépôts, des dépôts aux clients, ainsi que des clients aux dépôts :

$$A \subseteq (C \times D \times P) \cup (D \times D \times P) \cup (D \times C \times P).$$

Cette définition sous-entend que le réseau est composé d'arcs parallèles, un pour chaque type de conteneur, par direction. Le lecteur intéressé trouvera dans De Tocqueville (1987) une description détaillée de la pluralité des modes de transport et des divers types de conteneurs utilisés par une société européenne de transport

maritime et terrestre, en plus d'une étude statistique des données sur les différents mouvements possibles.

Définissons les données du modèle :

O_{ip} : offre au noeud i de conteneurs vides de type p, $i \in C$, $p \in P$;

D_{ip} : demande au noeud i de conteneurs vides de type p, $i \in C$, $p \in P$;

f_j : coût d'ouverture et d'opération (coût fixe) du dépôt j, $j \in D$;

c_{ijp} et c_{jip} : coûts unitaires de transport du conteneur vide de type p entre le client i et le dépôt j, $i \in C$, $j \in D$, $p \in P$;

s_{jkp} : coût unitaire de transport du conteneur vide de type p, entre les dépôts j et k, $j, k \in D$, $p \in P$;

puis les variables de décision :

$y = (y_j)$: vecteur d'ouverture des dépôts, $j \in D$;

$$y_j = \begin{cases} 1 & \text{dépôt } j \text{ ouvert} \\ 0 & \text{sinon} \end{cases}$$

$x = (x_{ijp}, x_{jip})$: vecteur des flots du conteneur vide de type p circulant entre le client i et le dépôt j, $i \in C$, $j \in D$, $p \in P$;

$w = (w_{jkp})$: vecteur des flots du conteneur vide de type p circulant entre les dépôts j et k, $j, k \in D$, $p \in P$.

Mentionnons que, dans le cas particulier du transport des conteneurs, les différents types de conteneurs correspondent au paramètre multiproduit du modèle.

I.2 Formulation mathématique

Le modèle de localisation-distribution multimode, multiproduit avec échanges interdépôts pour les coûts linéaires sans capacités s'écrit de la façon suivante :

$$(P)\quad \textit{Minimiser}\quad z = \sum_{j\in D} f_j y_j + \sum_{p\in P}\sum_{j\in D}\left[\sum_{i\in C}\left(c_{ijp}x_{ijp} + c_{jip}x_{jip}\right) + \sum_{k\in D} s_{kjp}w_{kjp}\right]$$

sujet à

$$\sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C, p\in P \qquad (1.1)$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C, p\in P \qquad (1.2)$$

$$x_{ijp} \le My_j \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.3)$$

$$x_{jip} \le My_j \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.4)$$

$$w_{jkp} \le My_j \qquad \forall\, j,k\in D, p\in P \qquad (1.5)$$

$$w_{kjp} \le My_j \qquad \forall\, k,j\in D, p\in P \qquad (1.6)$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D, p\in P \qquad (1.7)$$

$$x_{ijp} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.8)$$

$$x_{jip} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.9)$$

$$w_{jkp} \ge 0 \qquad \forall\, j,k\in D, p\in P \qquad (1.10)$$

$$y_j \in \{0,1\} \qquad \forall\, j\in D. \qquad (1.11)$$

Les contraintes (1.1) et (1.2) assurent que les disponibilités et demandes des clients sont satisfaites exactement, les contraintes (1.3) et (1.4) indiquent qu'il ne peut y avoir d'échanges entre les clients et les dépôts fermés (M est une constante positive, suffisamment grande), les contraintes (1.5) et (1.6) obligent les échanges

interdépôts à s'effectuer qu'entre dépôts ouverts, alors que les contraintes (1.7) représentent la conservation des flots aux dépôts. Les contraintes (1.8)-(1.11) sont celles de non-négativité et d'intégralité des variables.

Notons que la présence des équations de conservation de flot dans la formulation du problème confère à ce dernier une structure de réseau (voir figure 1). Ce sont ces contraintes qui maintiennent l'équilibre des conteneurs vides dans le réseau.

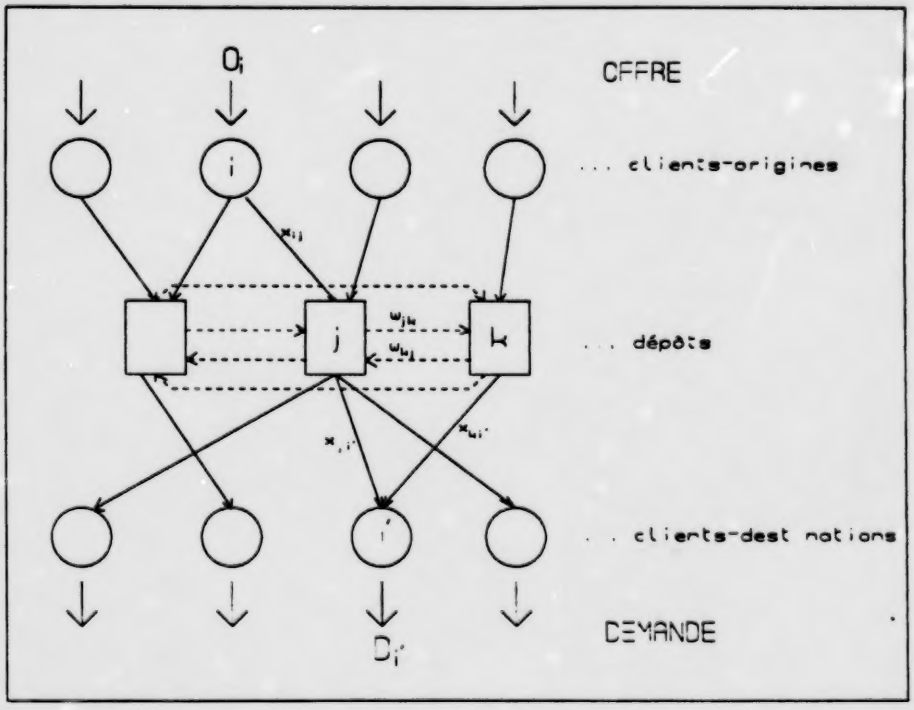


Figure 1. Structure de réseau présente dans le modèle.

Nous pouvons, sous certaines hypothèses, simplifier ce modèle. D'une part, les offres et les demandes peuvent remplacer la constante M dans les contraintes (1.3) et (1.4). Ceci a l'avantage de resserrer le domaine réalisable. D'autre part, lorsque les coûts de transport interdépôts satisfont l'inégalité du triangle :

$$s_{jkp} \leq s_{jlp} + s_{lkp} \qquad \forall\, j, k, l \in D,$$

Crainic, Dejax et Delorme (1986) ont démontré que les contraintes (1.5) et (1.6) sont redondantes et peuvent donc être éliminées de la formulation. Il s'agit d'une hypothèse réaliste et presque toujours vérifiée en pratique.

Après simplification, le modèle s'écrit :

$$(L)\quad \textit{Minimiser}\quad z = \sum_{j\in D} f_j y_j + \sum_{p\in P}\sum_{j\in D}\left[\sum_{i\in C}\left(c_{ijp}x_{ijp} + c_{jip}x_{jip}\right) + \sum_{k\in D} s_{kjp}w_{kjp}\right]$$

sujet à

$$\sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C, p\in P \qquad (1.1)$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C, p\in P \qquad (1.2)$$

$$x_{ijp} \le O_{ip} y_j \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.12)$$

$$x_{jip} \le D_{ip} y_j \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.13)$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D, p\in P \qquad (1.7)$$

$$x_{ijp} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.8)$$

$$x_{jip} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P \qquad (1.9)$$

$$w_{jkp} \ge 0 \qquad \forall\, j,k\in D, p\in P \qquad (1.10)$$

$$y_j \in \{0,1\} \qquad \forall\, j\in D. \qquad (1.11)$$

Dans la suite de ce mémoire, lorsque nous citerons le problème de localisation-distribution avec échanges interdépôts ou avec contraintes d'équilibrage ou lorsque nous parlerons tout simplement du problème originel ou initial, nous référerons à cette dernière formulation du problème.

I.3 Relaxations

Dans cette section, nous présentons deux problèmes obtenus à partir de la formulation précédente en relaxant certaines contraintes.

La première variante du problème de localisation-distribution avec contraintes d'équilibrage à laquelle nous nous intéressons est celle obtenue en appliquant à ce dernier la méthode des pénalités. La formulation est la suivante :

$$(PRL)\quad \textit{Minimiser}\quad z = \sum_{j\in D} f_j y_j + \sum_{p\in P}\sum_{j\in D}\left[\sum_{i\in C}\left(c_{ijp}x_{ijp} + c_{jip}x_{jip}\right) + \sum_{k\in D} s_{kjp}w_{kjp}\right] + \sum_{p\in P}\sum_{i\in C}\sum_{j\in D}\left[K_{ijp}(\min\{0, O_{ip}y_j - x_{ijp}\})^2 + K_{jip}(\min\{0, D_{ip}y_j - x_{jip}\})^2\right]$$

sujet à

$$\sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C, p\in P$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C, p\in P$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D, p\in P$$

$$x_{ijp} \geq 0 \qquad \forall\, i\in C, j\in D, p\in P$$

$$x_{jip} \geq 0 \qquad \forall\, i\in C, j\in D, p\in P$$

$$w_{jkp} \geq 0 \qquad \forall\, j,k\in D, p\in P$$

$$y_j \in \{0,1\} \qquad \forall\, j\in D$$

où K_{ijp} et K_{jip} sont des coefficients de pénalité strictement positifs.

La relaxation forte de (*PRL*) est un problème de multiflot sans capacités à coûts convexes (Crainic, Dejax et Delorme, 1986).

La deuxième variante que nous étudions, de manière plus approfondie, est la relaxation forte de (*L*), qui consiste à relaxer les contraintes d'intégralité sur les variables y_j. Elle s'énonce mathématiquement comme suit :

$$(SRL) \quad \textit{Minimiser} \quad z = \sum_{j \in D} f_j y_j + \sum_{p \in P} \sum_{j \in D} \left[\sum_{i \in C} \left(c_{ijp} x_{ijp} + c_{jip} x_{jip} \right) + \sum_{k \in D} s_{kjp} w_{kjp} \right]$$

sujet à

$$\sum_{j \in D} x_{ijp} = O_{ip} \qquad \forall\, i \in C, p \in P$$

$$\sum_{j \in D} x_{jip} = D_{ip} \qquad \forall\, i \in C, p \in P$$

$$x_{ijp} \leq O_{ip} y_j \qquad \forall\, i \in C, j \in D, p \in P \qquad (1.12)$$

$$x_{jip} \leq D_{ip} y_j \qquad \forall\, i \in C, j \in D, p \in P \qquad (1.13)$$

$$\sum_{i \in C} x_{ijp} + \sum_{k \in D} w_{kjp} - \sum_{i \in C} x_{jip} - \sum_{k \in D} w_{jkp} = 0 \qquad \forall\, j \in D, p \in P$$

$$x_{ijp} \geq 0 \qquad \forall\, i \in C, j \in D, p \in P$$

$$x_{jip} \geq 0 \qquad \forall\, i \in C, j \in D, p \in P$$

$$w_{jkp} \geq 0 \qquad \forall\, j, k \in D, p \in P$$

$$y_j \geq 0 \qquad \forall\, j \in D. \qquad (1.15)$$

Notons qu'il est inutile de spécifier la contrainte $y_j \leq 1$ pour chacun des dépôts, puisqu'elle est implicite par les contraintes de type (1.12) et (1.13). (*SRL*) est un problème de multiflot à coût minimum linéaire avec contrainte de capacité variable sur les variables x_{ijp} et x_{jip} (Crainic, Dejax et Delorme, 1986).

Mentionnons de plus, que la relaxation forte devrait générer une bonne borne inférieure sur la valeur optimale de la fonction économique du problème originel.

Ainsi, il pourrait être avantageux d'utiliser (*SRL*) pour le calcul des bornes dans un algorithme d'énumération implicite.

I.4 Stratégies de résolution

Nous formulons le problème de la localisation des dépôts des conteneurs vides et de la distribution de ces conteneurs parmi les clients comme un modèle d'optimisation linéaire, mixte de très grande taille. En effet, certaines variables sont entières (binaires), tandis que d'autres sont continues. De plus, la taille que peut atteindre ce problème devient rapidement énorme, à cause des contraintes de type (1.12) et (1.13). Il existe $2nml$ de telles contraintes. Ces deux facteurs rendent difficile la résolution d'un tel problème. Appliquer directement des techniques classiques de résolution de programmation mathématique devient une tâche complexe à cause du temps-machine considérable que requiert l'exécution de telles méthodes. Il faut donc adapter ces méthodes pour tirer avantage des caractéristiques et de la structure particulière du problème pour obtenir un algorithme efficace de résolution.

Tout d'abord, mentionnons une facette sous-jacente au problème, qui se révèle très utile dans les stratégies de résolution. Il s'agit de la structure de réseau qui émerge continuellement tout au long de l'analyse faite par Crainic, Dejax et Delorme (1986) du modèle lui-même ainsi que de tous les sous-problèmes : relaxations, dualisations, etc. Etant donné la richesse et la performance bien connues des algorithmes de réseau, cette structure s'avère très favorable aux développements algorithmiques. C'est donc vers cette avenue que les recherches ont été menées. Outre cette présente étude, deux autres méthodes ont été développées et programmées pour résoudre le problème de localisation-distribution avec exigences d'équilibrage. Nous précisons, dans ce qui suit, ces diverses approches.

La première (Hodgson, 1988) consiste en un algorithme d'énumération de type "branch-and-bound" basé sur les travaux de Khumawala (1972). Cet algorithme a l'avantage de fournir une solution optimale. Toutefois, en pratique, l'exécution du programme est très lente et son arrêt est provoqué avant même d'atteindre la solution

optimale. Le rendement de l'algorithme dépend des bornes calculées à chacun des noeuds du "branch-and-bound". Présentement, la relaxation faible de (L), obtenue en agrégeant les contraintes de type (1.12) et (1.13) dans la formulation du problème originel, sert au calcul de la borne inférieure de la méthode d'énumération. Cependant cette dernière n'est pas très bonne. De plus, à cause du temps de calcul considérable que nécessite cette méthode, la solution présentement générée se limite à une borne inférieure sur la valeur optimale du problème initial. Il serait donc avantageux d'utiliser la relaxation forte dans cet algorithme afin d'améliorer le calcul des bornes et ainsi accélérer la convergence de l'algorithme.

Quant à la deuxième méthode (Delorme, Crainic et Dejax, 1988), elle fait intervenir plusieurs procédures duales basées sur la méthode d'Erlenkotter (1978) faisant appel aux relations de complémentarité entre les formes primale et duale de la relaxation linéaire du problème. Il s'agit d'une approche de type "flip-flop" qui crée à la fois une borne inférieure et une borne supérieure sur la valeur optimale de la fonction objectif du problème originel. La première s'avère très bonne, tandis que la deuxième est moins intéressante, car son comportement est imprévisible.

En ce qui nous concerne, nous comptons également profiter de la structure de réseau présente dans le modèle, en considérant qu'il existe dans la littérature des algorithmes efficaces pour résoudre de tels problèmes. En particulier, les méthodes de relaxation primales-duales pour des problèmes de réseau semblent donner des résultats encourageants (Bertsekas, 1985; Bertsekas et Tseng, 1985, 1988; Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986). Dans cette perspective, nous envisageons deux stratégies de résolution pour aborder le problème de localisation-distribution avec équilibrage; la première consiste à traiter la relaxation forte de (PRL), alors que l'autre réside dans l'étude de (SRL). Nous examinons ces deux avenues de résolution de plus près dans le chapitre III et justifions notre choix qui s'est porté sur la deuxième. Nous décrivons, ici, le cheminement suivi pour développer une méthode de résolution à la relaxation forte du problème.

Tout d'abord, nous effectuons certaines manipulations sur la formulation de de la relaxation forte du problème de manière à éliminer l'hétérogénéité des variables.

Ceci nous permet de déduire un problème équivalent à (*SRL*) qui ne contient que des variables de flot et qui possède, lui aussi, une structure de réseau. Nous adoptons ensuite une approche de restriction, qui consiste à imposer des capacités sur certaines variables du problème transformé, restreignant ainsi son domaine réalisable. Nous définissons alors un sous-problème restreint dans lequel apparaît encore une fois une structure de réseau. Nous pouvons résoudre ce dernier par une adaptation de l'algorithme de relaxation primal-dual de Bertsekas, Hossein et Tseng (1987). Une telle approche génère donc une borne supérieure sur la valeur optimale de la fonction économique du problème initial. Nous espérons améliorer cette borne en résolvant plusieurs sous-problèmes restreints définis sur différentes portions du domaine réalisable.

La méthode que nous proposons permet donc de déterminer, de façon locale ou globale, différents domaines réalisables restreints, définissant les sous-problèmes restreints à résoudre. Une adaptation de l'algorithme de relaxation primal-dual s'insère alors dans notre procédure pour résoudre les sous-problèmes résultant de la restriction.

Nous espérons, par cette approche, obtenir une solution entière. Sinon, celle-ci pourrait s'avérer une très bonne solution initiale pour un algorithme d'énumération. Selon cette dernière éventualité, la borne supérieure produite par notre méthode pourrait alors être utilisée conjointement à la borne inférieure fournie par la méthode duale dans un algorithme d'énumération, tel le "branch-and-bound".

Avant de présenter notre procédure, nous rappelons les méthodes de relaxation de type primal-dual pour les problèmes de flot à coût minimum linéaire et convexe avec capacités (Bertsekas, 1985; Bertsekas et Tseng, 1985, 1988; Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986; Tseng et Bertsekas, 1987).

CHAPITRE II

METHODES DE RELAXATION PRIMALES-DUALES

Dans ce chapitre, nous présentons deux méthodes de relaxation de type primal-dual pour résoudre les problèmes de flot à coût minimum linéaire (Bertsekas, 1985; Bertsekas et Tseng, 1985, 1988; Tseng, 1986; Tseng et Bertsekas, 1987) et convexe (Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986) avec capacités. Tout d'abord, qu'est-ce qu'une méthode de relaxation? Puis, qu'est-ce qu'une méthode primale-duale?

La philosophie des méthodes de relaxation (Geoffrion, 1969) consiste à résoudre une version relaxée du problème en ne prenant pas en considération certaines contraintes. Si la solution résultante ne satisfait pas les contraintes ignorées, alors on génère une ou plusieurs de ces contraintes que l'on ajoute au problème "relaxé". puis on résout de nouveau. On continue ainsi jusqu'à ce que la solution satisfasse toutes les contraintes mises de côté, c'est-à-dire jusqu'à ce que l'on trouve une solution optimale au problème donné.

Par ailleurs, le principe des méthodes primales-duales permet de travailler simultanément sur les formulations primale et duale du même problème. Pour mieux saisir ce concept, considérons, par exemple, une méthode primale-duale bien connue traitant du problème de flot à coût minimum avec capacités dans un réseau. Il s'agit

de l'algorithme des arcs *non–conformes*† (Ford et Fulkerson, 1961). L'algorithme débute avec un flot, réalisable ou non, satisfaisant les équations de conservation de flot, et un ensemble de variables duales, appelées prix ou potentiels. La procédure vise à satisfaire des conditions de conformité qui sont en fait les conditions de complémentarité. Cela se fait en modifiant itérativement, soit le flot sur les arcs par la recherche d'un chemin, soit les prix aux noeuds par la recherche d'une coupe. L'algorithme réduit de façon monotone la non-réalisabilité ou les déviations aux écarts complémentaires des arcs non-conformes, arcs pour lesquels les conditions de complémentarité sont violées.

Les procédures que nous étudions dans ce chapitre sont une combinaison de ces deux approches. Il s'agit de méthodes de type primal-dual qui s'apparentent à celle des arcs non-conformes, à la différence toutefois qu'elles respectent les conditions de complémentarité et cherchent à satisfaire les équations de conservation de flot. Contrairement à ce que l'on retrouve dans la littérature sur les méthodes primales-duales (à cet effet, mentionnons Shapiro, 1977), les promoteurs des méthodes de relaxation primales-duales (Bertsekas et Tseng, 1985; Bertsekas, Hossein et Tseng, 1987) utilisent la théorie de la programmation monotropique pour développer leurs algorithmes. L'intérêt de leurs recherches provient du fait que cette théorie permet de développer des algorithmes d'optimisation pour résoudre le problème de réseau dans lesquels les flots ou les prix sont améliorés dans un seul circuit ou une seule coupe à la fois (Rockafellar, 1984), d'où le sens d'une méthode de relaxation. Le lecteur avide d'en connaître davantage sur la programmation monotropique peut satisfaire sa curiosité en lisant la dernière référence citée.

Dans ce chapitre, nous décrivons la méthode de relaxation primale-duale qui résout le problème de flot à coût minimum dans un réseau. Nous décrivons d'abord l'algorithme pour les coûts linéaires (Bertsekas, 1985; Bertsekas et Tseng, 1985, 1988; Tseng, 1986; Tseng et Bertsekas, 1987). C'est le cas le plus simple, il nous permet de nous familiariser avec la méthode. Nous abordons ensuite le cas plus

† Cette méthode est plus connue sous son nom anglais *"out-of-kilter"*.

général des coûts convexes (Bertsekas, Hossein et Tseng, 1985, 1987; Tseng, 1986).

Contrairement aux auteurs de la méthode qui exposent le problème dans le contexte de la programmation monotropique, nous avons choisi de présenter le problème dans un contexte plus académique, c'est-à-dire du point de vue de la programmation mathématique linéaire standard (Luenberger, 1973). Dans le cas des coûts linéaires, aucun changement au développement théorique n'est requis. En revanche, pour le cas des coûts convexes, de légères modifications s'imposent. Il est alors nécessaire de démontrer l'équivalence entre les hypothèses auxquelles nous avons recours et celles utilisées par Bertsekas, Hossein et Tseng (1987) afin de justifier l'algorithme de relaxation.

II.1 Coûts linéaires

Dans cette section, nous formulons le problème de flot à coût minimum linéaire avec capacités dans un réseau, son dual, ainsi que les conditions d'optimalité relatives à ces deux problèmes. Nous introduisons les notations et définitions utiles à la description de la méthode qui serviront également pour le cas convexe. Nous terminons par la présentation de l'algorithme.

II.1.1 Problème

Considérons un réseau $G = (N, A)$ où N est l'ensemble des noeuds et A est l'ensemble des arcs (i, j).

Dénotons par :

c_{ij} : le coût associé à l'arc (i, j)

x_{ij} : le flot circulant sur l'arc (i, j)

l_{ij} : la borne inférieure sur le flot de l'arc (i, j)

u_{ij} : la borne supérieure sur le flot de l'arc (i, j).

Le problème de flot à coût minimum linéaire avec capacités s'écrit :

$$(P\,1) \qquad \textit{Minimiser } z = \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{2.1}$$

sujet à

$$\sum_{j\in N} x_{ji} - \sum_{j\in N} x_{ij} = 0 \qquad \forall\, i\in N \tag{2.2}$$

$$0 \le l_{ij} \le x_{ij} \le u_{ij} \qquad \forall\, (i,j)\in A. \tag{2.3}$$

Les contraintes (2.2) correspondent aux équations de conservation de flot aux noeuds et les contraintes (2.3) imposent des capacités sur la quantité de flot pouvant circuler sur chacun des arcs $(i, j)\in A$.

Un flot $x = (x_{ij})\ \forall\, (i, j)\in A$ est dit réalisable pour le problème $(P\,1)$ s'il satisfait (2.2) et (2.3); il est dit optimal si, en plus, il satisfait (2.1). Nous supposons tout au long de cette section qu'il existe un flot réalisable pour $(P\,1)$.

II.1.2 Problème dual et conditions d'optimalité

Le problème dual de $(P\,1)$ se formule comme suit :

$$(D\,1) \qquad \textit{Maximiser } \Psi = \sum_{(i,j)\in A} \lambda_{ij} l_{ij} - \sum_{(i,j)\in A} \gamma_{ij} u_{ij}$$

sujet à

$$-\pi_j + \pi_i + \lambda_{ij} - \gamma_{ij} \le c_{ij} \qquad \forall\, (i,j)\in A$$

$$\lambda_{ij} \ge 0 \qquad \forall\, (i,j)\in A$$

$$\gamma_{ij} \ge 0 \qquad \forall\, (i,j)\in A$$

$$\pi_i \text{ quelconque} \qquad \forall\, i\in N.$$

Les variables λ_{ij} et γ_{ij} sont associées aux contraintes primales $x_{ij} \geq l_{ij}$ et $-x_{ij} \geq -u_{ij}$, tandis que les variables duales $-\pi_i$ correspondent aux équations de conservation de flot du problème primal. π_i est appellé le *prix* au noeud i.

Dans ce qui suit, lorsque nous faisons référence à un vecteur, nous omettons les indices. Par exemple, π correspond au vecteur, ayant comme éléments $\pi_i \ \forall\, i \in N$.

Les *conditions d'optimalité* relatives aux problèmes primal et dual s'énoncent ainsi :

$(x^*, \pi^*, \gamma^*, \lambda^*)$ est une solution optimale primale-duale pour $(P\,1)$ et $(D\,1)$ ssi

$i)$ x^* est un flot réalisable pour $(P\,1)$

$ii)$ $(\pi^*, \gamma^*, \lambda^*)$ est une solution réalisable pour $(D\,1)$

$iii)$ $\lambda_{ij}^* (x_{ij}^* - l_{ij}) = 0 \qquad \forall\, (i, j) \in A$

$iv)$ $\gamma_{ij}^* (u_{ij} - x_{ij}^*) = 0 \qquad \forall\, (i, j) \in A$

$v)$ $(c_{ij} - \pi_i^* + \pi_j^* - \lambda_{ij}^* + \gamma_{ij}^*)\, x_{ij}^* = 0 \qquad \forall\, (i, j) \in A.$

Il est possible d'éliminer les variables λ_{ij} et γ_{ij} de la formulation du problème dual en remarquant que les conditions ci-dessus sont équivalentes aux conditions suivantes (Lawler, 1976; Shapiro, 1977) :

$i')$ $\sum_{\substack{j \in N \\ (j,\, i) \in A}} x_{ji}^* - \sum_{\substack{j \in N \\ (i,\, j) \in A}} x_{ij}^* = 0 \qquad \forall\, i \in N$

$ii')$ $c_{ij} - \pi_i^* + \pi_j^* > 0 \ \Rightarrow \ x_{ij}^* = l_{ij} \qquad \forall\, (i, j) \in A$

$iii')$ $c_{ij} - \pi_i^* + \pi_j^* = 0 \ \Rightarrow \ l_{ij} \leq x_{ij}^* \leq u_{ij} \qquad \forall\, (i, j) \in A$

$iv')$ $c_{ij} - \pi_i^* + \pi_j^* < 0 \ \Rightarrow \ x_{ij}^* = u_{ij} \qquad \forall\, (i, j) \in A$

où les conditions ii' à iv' correspondent aux *conditions de complémentarité*, que nous notons (CC).

Le problème dual ($D\,1$) peut donc se retranscrire sous la forme suivante :

$$(D\,1') \qquad \underset{\pi}{\textit{Maximiser}} \quad q(\pi)$$

où la fonction duale q est donnée par

$$q(\pi) = \sum_{(i,\,j)\in A} g_{ij}(\pi_i - \pi_j)$$

où

$$g_{ij}(\pi_i - \pi_j) = \underset{l_{ij} \le x_{ij} \le u_{ij}}{\textit{infimum}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\}. \tag{2.4}$$

Les fonctions de coût primale et duale associées à un arc (i, j) sont tracées dans les figures 2.1 et 2.2.

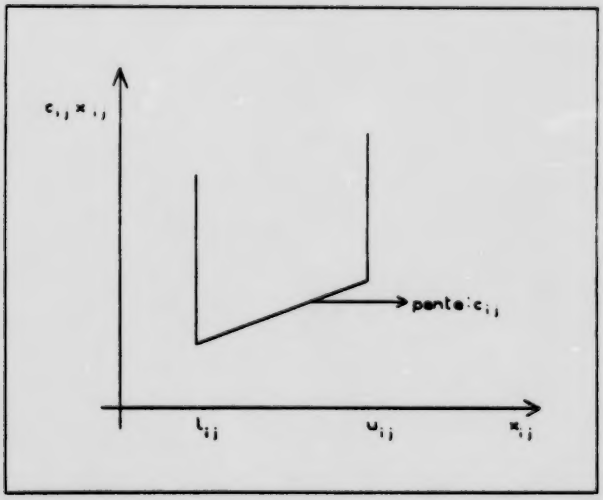


Figure 2.1. Graphe de la fonction linéaire de coût primale associée à un arc (i, j).

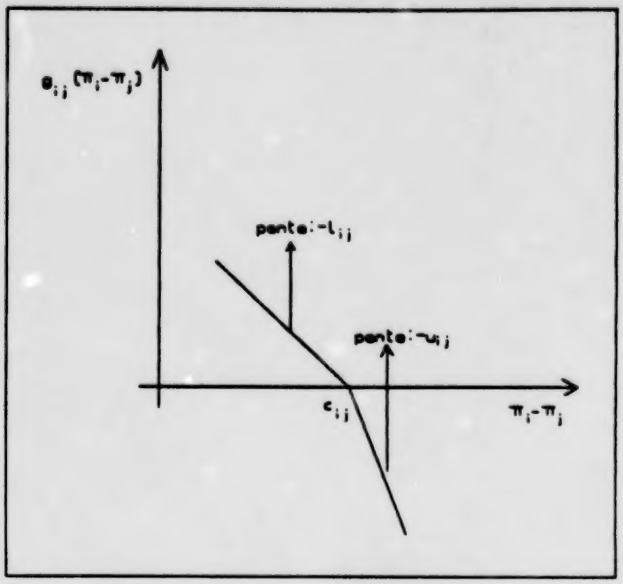


Figure 2.2. Graphe de la fonction linéaire de coût duale associée à un arc (i, j).

Soulignons que cette formulation, générée via la dualité de la programmation mathématique linéaire standard (Luenberger, 1973), peut aussi être obtenue à partir de la dualité Lagrangienne (Geoffrion, 1971) ou la dualité convexe (Rockafellar, 1970). Dans cette dernière référence, g_{ij} est appelée la *fonction convexe conjuguée* de f_{ij}. En outre, dans le cas linéaire, le développement du problème dual, dans un contexte comme dans l'autre, ne nécessite aucune hypothèse particulière. Nous pouvons donc utiliser directement la théorie qu'offre la programmation monotropique pour présenter l'algorithme de relaxation. Tout d'abord, introduisons la terminologie.

II.1.3 Notations et définitions

Soit un vecteur de prix π donné, la *tension* sur l'arc (i, j) est définie par :

$$t_{ij} = \pi_i - \pi_j \quad \forall\, (i, j) \in A.$$

Le vecteur tension t correspond au vecteur de prix π. Puisque la fonction de coût dual dépend du vecteur de prix seulement via le vecteur de tension correspondant t, bien souvent, nous ne ferons aucune distinction entre π et t dans ce qui suit.

Pour un vecteur de prix π quelconque, un arc (i, j) est dit :

$$\begin{array}{lll} \textit{inactif} & \text{si} & t_{ij} < c_{ij}, \\ \textit{libre} & \text{si} & t_{ij} = c_{ij} \quad \text{et} \\ \textit{actif} & \text{si} & t_{ij} > c_{ij} \end{array} \qquad \forall\, (i, j) \in A.$$

Pour un vecteur de flot x quelconque, le *déficit* au noeud i, d_i est défini par :

$$d_i = \sum_{\substack{j \in N \\ (i, j) \in A}} x_{ij} - \sum_{\substack{j \in N \\ (j, i) \in A}} x_{ji} \qquad \forall\, i \in N.$$

En d'autres mots, le déficit au noeud i représente la différence entre la somme totale de flot exporté et la somme totale de flot importé au noeud i.

Le *déficit total* pour le vecteur x est défini par $\sum_{i \in N} |d_i|$.

Le déficit total est une mesure de de la "réalisabilité" du vecteur x à satisfaire le système des équations de conservation de flot.

Une manière équivalente d'écrire les (CC) est :

$$\begin{array}{lll} i'') & x_{ij} = l_{ij} & \text{pour tous les arcs inactifs } (i, j) \\ ii'') & l_{ij} \le x_{ij} \le u_{ij} & \text{pour tous les arcs libres } (i, j) \\ iii'') & x_{ij} = u_{ij} & \text{pour tous les arcs actifs } (i, j) \end{array} \qquad (CC')$$

Conséquemment, les conditions d'optimalité énoncées précédemment deviennent :

$$x \text{ et } \pi \text{ satisfont les } (CC') \quad \text{et} \quad d_i = 0 \quad \forall\, i \in N.$$

Soit (x, π) une paire primale-duale de vecteurs satisfaisant les (CC'). Une séquence de noeuds $\{n_1, n_2, \cdots, n_k\}$ est une *chaîne d'augmentation de flot*, que nous dénotons *CAF*, si $d_{n_1} < 0$, $d_{n_k} > 0$ et pour $m = 1, 2, \cdots, k-1$, ou bien il existe un arc libre (n_m, n_{m+1}) avec $x_{n_m n_{m+1}} < u_{n_m n_{m+1}}$, ou bien un arc libre (n_{m+1}, n_m) avec $x_{n_{m+1} n_m} > l_{n_{m+1} n_m}$.

La *capacité* de la chaîne δ est définie par :

$$\delta = \min\{ -d_{n_1}, d_{n_k}, \delta_1, \cdots, \delta_{k-1} \}$$

où

$$\delta_m = \begin{cases} u_{ij} - x_{ij} & \text{si } (i,j) \in CAF \text{ et } (i,j) \text{ est orienté de } n_1 \text{ vers } n_k \\ x_{ij} - l_{ij} & \text{si } (i,j) \in CAF \text{ et } (i,j) \text{ est orienté de } n_k \text{ vers } n_1. \end{cases}$$

L'ensemble des sommets *adjacents (voisins)* au sommet i $\Gamma(i)$ est défini par :

$$\Gamma(i) = \{ j \in N \mid (i, j) \in A \} \cup \{ j \in N \mid (j, i) \in A \} \quad \forall i \in N.$$

L'approche proposée par Bertsekas et Tseng (1985) est basée sur des augmentations itératives de la fonction duale. Le vecteur de prix π est mis à jour en maintenant simultanément les (CC') satisfaites par rapport aux vecteurs flot et prix. L'algorithme se termine lorsque x devient primal-réalisable, c'est-à-dire lorsque le déficit à chacun des noeuds égal zéro. Le principal attrait de cette méthode comparativement aux autres méthodes primales-duales provient du choix très simple des directions de montée. En effet, étant donné un vecteur de prix π, un noeud i avec un déficit non-nul est choisi et une montée le long de la coordonnée π_i est tentée. Si une telle opération de même qu'une diminution du déficit total sont impossibles, un noeud adjacent au noeud i, disons le noeud j, est choisi et une montée le long de la somme des coordonnées i et j est essayée. Si cela demeure impossible, de même qu'une augmentation de flot, alors le processus se poursuit en choisissant un noeud adjacent

à i ou à j. En pratique, Bertsekas et Tseng (1988) ont constaté que la plupart des directions de montée correspondent à des directions n'impliquant qu'une seule coordonnée.

Caractérisons maintenant la notion de direction de montée duale. Pour ce faire, démontrons la proposition de Bertsekas et Tseng (1985) :

Proposition 2.1

Pour un vecteur de prix π donné et un sous-ensemble connexe de noeuds S quelconque de N,

$$C(S, \pi) = q'(\pi; \mu(S))$$

où

$$C(S,\pi) = - \sum_{\substack{(i,j)\in(\bar{S},S) \\ \textit{inactif ou libre}}} l_{ij} - \sum_{\substack{(i,j)\in(\bar{S},S) \\ \textit{actif}}} u_{ij} + \sum_{\substack{(i,j)\in(S,\bar{S}) \\ \textit{inactif}}} l_{ij} + \sum_{\substack{(i,j)\in(S,\bar{S}) \\ \textit{actif ou libre}}} u_{ij}$$

$$\mu_i(S) = \begin{cases} -1 & \text{si } i \in S \\ 0 & \text{si } i \in \bar{S} \quad (\bar{S} = N - S) \end{cases}$$

$q'(\pi; \mu(S))$ est la dérivée directionnelle de la fonction duale q *en* π dans la direction $\mu(S)$.

preuve:

Par définition de la dérivée directionnelle, nous avons :

$$q'(\pi; \mu(S)) = \lim_{\Delta \downarrow 0} \frac{q(\pi + \Delta\mu(S)) - q(\pi)}{\Delta}.$$

En substituant (2.4),

$$q'(\pi; \mu(S)) = \lim_{\Delta \downarrow 0} \left\{ \sum_{(i,j)\in A} \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \Delta\mu_i + \pi_j - \Delta\mu_j)\, x_{ij} \right\} \right.$$
$$\left. - \sum_{(i,j)\in A} \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right\} / \Delta$$

$$= \lim_{\Delta \downarrow 0} \left\{ \sum_{(i,j) \in (S,\bar{S})} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \Delta + \pi_j - \Delta)\, x_{ij} \right\} - \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right] \right.$$

$$+ \sum_{(i,j) \in (S,\bar{S})} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \Delta + \pi_j)\, x_{ij} \right\} - \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right]$$

$$+ \sum_{(i,j) \in (\bar{S},S)} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j - \Delta)\, x_{ij} \right\} - \inf_{x_{ij}} \left\{ (c_{ij} - \pi + \pi_j)\, x_{ij} \right\} \right]$$

$$\left. + \sum_{(i,j) \in (\bar{S},\bar{S})} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} - \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right] \right\} / \Delta$$

$$= \lim_{\Delta \downarrow 0} \left\{ \sum_{(i,j) \in (S,\bar{S})} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} + \inf_{x_{ij}} \{ \Delta\, x_{ij} \} + \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right] \right.$$

$$+ \sum_{(i,j) \in (\bar{S},S)} \left[\inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} + \inf_{x_{ij}} \{ -\Delta\, x_{ij} \} \right.$$

$$\left. \left. + \inf_{x_{ij}} \left\{ (c_{ij} - \pi_i + \pi_j)\, x_{ij} \right\} \right] \right\} / \Delta$$

$$= \lim_{\Delta \downarrow 0} \frac{\Delta \left\{ \sum_{(i,j) \in (S,\bar{S})} \inf_{x_{ij}} \{ x_{ij} \} + \sum_{(i,j) \in (\bar{S},S)} \inf_{x_{ij}} \{ -x_{ij} \} \right\}}{\Delta}$$

en prenant l'infimum de x_{ij} sur $[l_{ij}, u_{ij}]$, nous obtenons

$$= \sum_{\substack{(i,j) \in (S,\bar{S}) \\ \text{inactif ou libre}}} l_{ij} + \sum_{\substack{(i,j) \in (S,\bar{S}) \\ \text{actif}}} u_{ij} - \sum_{\substack{(i,j) \in (\bar{S},S) \\ \text{actif ou libre}}} u_{ij} - \sum_{\substack{(i,j) \in (\bar{S},S) \\ \text{inactif}}} l_{ij}$$

d'où,

$$q'(\pi, \mu(S)) = C(S, \pi)\,. \square$$

Un corollaire immédiat est le suivant :

si $C(S, \pi) > 0$, alors $\mu(S)$ est une direction de montée de la fonction duale q le long de π.

Ainsi, une modification du vecteur de prix dans la direction $\mu(S)$ correspond à diminuer les prix de tous les noeuds appartenant à S d'une même quantité tout en maintenant inchangé le prix de tous les autres noeuds. Tseng (1986) démontre que cette quantité, dénotée θ, est la plus petite valeur pour laquelle π peut être modifié sans violer les (CC'). Elle est donnée par :

$$\theta = \min \left\{ \min_{(i,j)\in S_1} (t_{ij} - c_{ij}),\ \min_{(i,j)\in S_2} (c_{ij} - t_{ij}) \right\} \tag{2.5}$$

où

$S_1 = \{ (i, j)\in(S,\bar{S}) \mid (i,j)$ est actif $\}$

$S_2 = \{ (i, j)\in(\bar{S},S) \mid (i,j)$ est inactif $\}$.

En d'autres mots, θ est la valeur pour laquelle un nouvel arc incident à S devient libre. Si $S_1 \cup S_2 = \varnothing$, la convention est que $\theta = \infty$.

II.1.4 Algorithme

L'algorithme de relaxation proposé par Bertsekas et Tseng (1988) maintient les (CC') à chacune des itérations tout en essayant de satisfaire la condition de réalisabilité (2.2). Le but de la procédure consiste donc à éliminer les déficits aux noeuds. Le processus commence avec un noeud ayant un déficit non-nul et vérifie s'il est possible, en modifiant les prix aux noeuds, d'augmenter la valeur de la fonction duale. Sinon, la procédure utilise la règle d'étiquetage de Ford et Fulkerson (1961) pour construire une chaîne d'augmentation de flot ou une coupe associée à une direction de montée. L'algorithme se termine lorsque tous les déficits sont nuls.

Dans le texte suivant, nous exposons une itération traitant le cas d'un noeud avec un déficit positif. Soulignons qu'une itération faite sur un noeud ayant un déficit négatif se construit de la même manière. Les détails d'une telle itération sont fournis à l'appendice B.

Itération de l'algorithme de relaxation (coûts linéaires)

Etape 0. Soit une paire primale-duale de vecteurs (x,π) satisfaisant les (CC').

Etape 1. Si $d_i = 0 \ \forall i \in N$ alors *STOP*.

Sinon faire : $M = \varnothing$

$S = \varnothing$

où

M = ensemble des sommets marqués non-examinés

S = ensemble des sommets marqués et examinés.

Choisir un sommet i tel que $d_i > 0$.

Faire : $M = \{i\}$

Aller à l'étape 2.

Etape 2. Choisir un sommet $k \in M$.

Faire : $S \leftarrow S \cup \{k\}$

$M \leftarrow M - \{k\}$.

Aller à l'étape 3.

Etape 3. Examiner le sommet k en marquant tous les sommets non-marqués $j \in \Gamma(k)$ et en leur donnant l'étiquette k de la manière suivante :

$$M \leftarrow M \cup \{j\} \quad \text{si} \quad \begin{cases} (k,j) \in A \mid x_{kj} > l_{kj} \text{ et } (k,j) \text{ libre} \\ \text{ou} \\ (j,k) \in A \mid x_{jk} < u_{jk} \text{ et } (j,k) \text{ libre.} \end{cases}$$

Si $C(S,\pi) > 0$ aller à l'étape 5.

Sinon,

s'il existe un sommet $j \in M$ étiqueté à partir de k *t.q.* $d_j < 0$ aller à l'étape 4.

Sinon retourner à l'étape 2.

Etape 4. Retracer la *CAF* reliant les sommets j et i (j est le sommet marqué avec $d_j < 0$). Soit δ la capacité de la chaîne. Augmenter de δ le flot sur les arcs orientés de j vers i et enlever δ unités de flot sur les arcs orientés de i vers j. Ajuster les déficits aux extrémités de la chaîne. Passer à l'itération suivante.

Etape 5. Soit $\bar{S} = N - S$, ($\bar{S} \neq \emptyset$ car $d_i > 0$ et $d_j \geq 0 \ \forall j \in S$ et $\sum_{i \in N} d_i = 0$).

Faire : $x_{kj} \leftarrow l_{kj} \quad \forall (k,j) \in (S,\bar{S})$ libre et $k \in M$

$x_{jk} \leftarrow u_{jk} \quad \forall (j,k) \in (S,\bar{S})$ libre et $k \in M$.

Faire $p_k \leftarrow p_k - \theta \quad \forall k \in S$ où θ est donné par (2.5).

Passer à l'itération suivante.

II.2 Coûts convexes

Dans cette section, nous présentons l'algorithme proposé par Bertsekas, Hossein et Tseng (1987) pour résoudre le problème de flot à coût minimum avec coûts convexes et capacités. Toutefois, nous formulons le problème dans un contexte théorique différent de celui de la programmation monotropique (Rockafellar, 1984) utilisé par ces auteurs. Etant donné notre formation académique, nous préférons avoir recours à la théorie de la dualité lagrangienne (Geoffrion, 1971). L'utilisation de cette théorie nous oblige à introduire des hypothèses différentes de celles que nous retrouvons dans Bertsekas, Hossein et Tseng (1985). De nouvelles notations et définitions s'ajoutent à celles de la section précédente. Enfin, nous montrons l'équivalence entre nos hypothèses et celles utilisées par les auteurs de la méthode de relaxation afin de justifier leur algorithme.

II.2.1 Problème

Considérons le cas plus général du problème de flot à coût minimum, où le coût c_{ij}, associé à chaque arc (i,j), est remplacé par une fonction de coût f_{ij}. Afin de faciliter les développements théoriques, chacune des fonctions f_{ij} est supposée convexe. La formulation du problème de flot à coût minimum convexe est la suivante :

$$(P2) \qquad \textit{Minimiser } f(x) = \sum_{(i,j)\in A} f_{ij}(x_{ij})$$

sujet à

$$\sum_{j\in N} x_{ji} - \sum_{j\in N} x_{ij} = 0 \qquad \forall\, i \in N$$

$$l_{ij} \le x_{ij} \le u_{ij} \qquad \forall\, (i,j) \in A.$$

Nous supposons que $(P2)$ possède une solution réalisable.

Le programme d'optimisation $(P2)$ est un problème de programmation convexe séparable avec contraintes linéaires. Il fait donc partie de la grande classe des programmes monotropiques (Rockafellar, 1984). C'est d'ailleurs dans le contexte de la programmation monotropique que Bertsekas, Hossein et Tseng (1987) ont choisi de développer leur méthode. Toutefois, pour simplifier l'exposé, nous choisissons de présenter le problème dans le cadre classique primal-dual (Geoffrion, 1971). Pour ce faire, de nouvelles hypothèses doivent être considérées.

Nous supposons que chacune des fonctions f_{ij} est continue et que le problème primal est stable (cf. appendice A). Ces conjectures nous permettent donc d'utiliser la théorie duale lagrangienne (Geoffrion, 1971).

II.2.2 Problème dual

Soit $\pi = [\pi_1, \pi_2, \cdots, \pi_n]$ un vecteur des multiplicateurs de Lagrange associé aux contraintes de conservation de flot. Le problème dual de $(P2)$ s'écrit :

$$(D2) \qquad \underset{\pi}{Maximiser}\ q(\pi) = \sum_{(i,j)\in A} \underset{l_{ij}\le x_{ij}\le u_{ij}}{infimum} \left\{ f_{ij}(x_{ij}) - (\pi_i - \pi_j)\, x_{ij} \right\}.$$

Cette formulation est équivalente à celle obtenue par Bertsekas, Hossein et Tseng (1987) en utilisant la dualité de Rockafellar (1971) basée sur les fonctions convexes conjuguées. En effet, un réarrangement des termes engendre le problème dual suivant :

$$(D2') \qquad \underset{\pi}{-Minimiser}\ -q(\pi) = \sum_{(i,j)\in A} g_{ij}(\pi_i - \pi_j)$$

où

$$g_{ij}(\pi_i - \pi_j) = \underset{l_{ij}\le x_{ij}\le u_{ij}}{supremum} \left\{ -f_{ij}(x_{ij}) + (\pi_i - \pi_j)\, x_{ij} \right\}. \qquad (2.6)$$

Dans le contexte de la dualité de Rockafellar (1971), g_{ij} est appelée la *fonction convexe conjuguée* de la fonction de coût primal f_{ij}. Les fonctions de coût primale et duale associées à un arc (i, j) sont tracées dans les figures 2.3 et 2.4.

$(P2)$ et $(D2')$ constituent une paire de problèmes monotropiques (Rockafellar, 1984). Ces programmes possèdent la propriété importante qui suit : à chaque point non-optimal, il est possible de trouver une direction de descente parmi un nombre fini de directions. L'idée maîtresse de l'algorithme de relaxation repose sur cette notion. Avant de le présenter, nous altérons certaines définitions et notations déjà introduites et en intégrons de nouvelles.

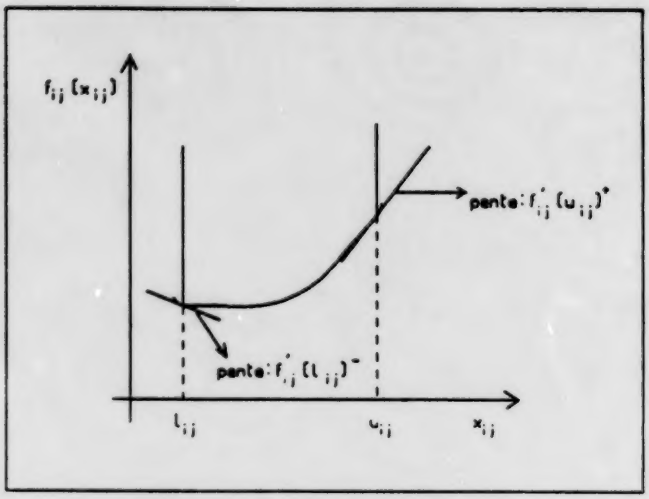


Figure 2.3. Graphe de la fonction convexe de coût primale associée à un arc (i, j).

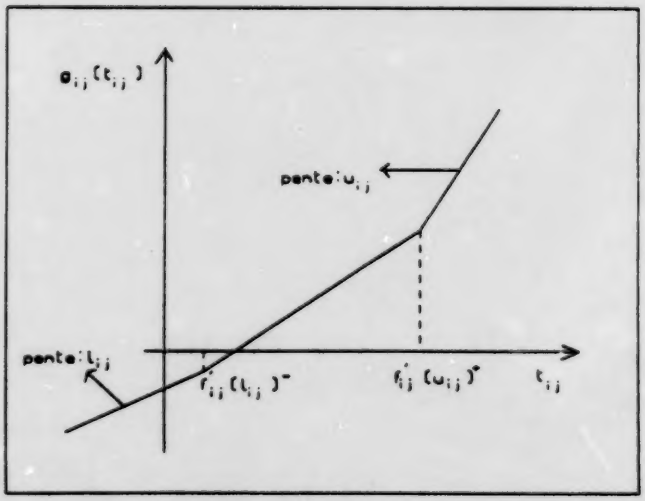


Figure 2.4. Graphe de la fonction convexe de coût duale associée à un arc (i, j).

II.2.3 Notations et définitions

Pour un $\varepsilon > 0$ donné, $x \in R^{|A|}$, $\pi \in R^{|N|}$, x et π satisfont les ε–conditions de complémentarité, dénotées (ε–*CC*), si

$$f'_{ij}(x_{ij})^- - \varepsilon \le t_{ij} \le f'_{ij}(x_{ij})^+ + \varepsilon \qquad \forall (i,j) \in A \tag{2.7}$$

où

$f'_{ij}(\xi)^+$: dérivée à droite au point ξ de f_{ij}

$f'_{ij}(\xi)^-$: dérivée à gauche au point ξ de f_{ij}

t_{ij} : tension associée à l'arc (i, j) (cf. section II.1.3).

Se rapportant au cas linéaire, nous pouvons faire un rapprochement entre ces nouvelles définitions et celles de la section II.1.3. Les (ε–*CC*) peuvent être interprétées comme les (*CC*) dans l'algorithme de relaxation. Ainsi, de façon intuitive, si x est primal-réalisable et (x, π) satisfont les (ε–*CC*), alors x et π devraient être à une distance ε de l'optimum.

Pour un π donné, (2.7) définit les bornes inférieures et supérieures sur les variables de flot, appelées les *bornes* –ε. Elles sont données par :

$$\begin{aligned} l_{ij}^{\varepsilon} &= \inf\{\, \xi \mid f'_{ij}(\xi)^+ \ge t_{ij} - \varepsilon \,\} \\ u_{ij}^{\varepsilon} &= \sup\{\, \xi \mid f'_{ij}(\xi)^- \le t_{ij} + \varepsilon \,\} \end{aligned} \qquad \forall (i,j) \in A. \tag{2.8}$$

Ainsi, x et π satisfont les (ε–*CC*) est équivalent à :

$$x_{ij} \in [l_{ij}^{\varepsilon}, u_{ij}^{\varepsilon}] \qquad \forall (i,j) \in A. \tag{ε–CC'}$$

De (2.8), on peut déduire la propriété suivante :

$$\varepsilon \ge \varepsilon' \;\Rightarrow\; l_{ij}^{\varepsilon} \le l_{ij}^{\varepsilon'} \text{ et } u_{ij}^{\varepsilon} \ge u_{ij}^{\varepsilon'} \qquad \forall (i,j) \in A. \tag{2.9}$$

Dans le cas convexe, nous démontrons une proposition analogue (Bertsekas, Hossein et Tseng, 1987) à la proposition 2.1 pour le cas linéaire :

Proposition 2.2

Pour $\varepsilon \geq 0$, un vecteur de prix π donné et un sous-ensemble de noeuds connexe S quelconque de N,

$$C_\varepsilon(S,\pi) \geq -q'(\pi;\ \mu(S))$$

où

$$C_\varepsilon(S,\pi) = \sum_{(i,j)\in(S,\bar{S})} l_{ij}^\varepsilon - \sum_{(i,j)\in(\bar{S},S)} u_{ij}^\varepsilon$$

$$\mu_i(S) = \begin{cases} -1 & \text{si } i \in S \\ 0 & \text{si } i \in \bar{S} \end{cases}$$

$q'(\pi;\ \mu(S))$ est la dérivée directionnelle de la fonction duale q en π dans la direction $\mu(S)$.

preuve:

Par définition de la dérivée directionnelle, nous avons :

$$q'(\pi;\ \mu(S)) = \lim_{\Delta \downarrow 0} \frac{q(\pi + \Delta\mu(S)) - q(\pi)}{\Delta}.$$

Par un raisonnement similaire à celui fait dans la preuve de la proposition 2.1, nous obtenons

$$q'(\pi;\ \mu(S)) = \lim_{\Delta \downarrow 0} \frac{\Delta \left\{ \sum_{(i,j)\in(S,\bar{S})} \sup_{x_{ij}} \{-x_{ij}\} + \sum_{(i,j)\in(\bar{S},S)} \sup_{x_{ij}} x_{ij} \right\}}{\Delta}.$$

En prenant le supremum sur $[l_{ij},\ u_{ij}]$,

$$q'(\pi;\ \mu(S)) = \sum_{(i,j)\in(S,\bar{S})} -l_{ij}^\varepsilon + \sum_{(i,j)\in(\bar{S},S)} u_{ij}^\varepsilon \qquad \text{pour } \varepsilon = 0.$$

D'où,

$$q'(\pi; \mu(S)) = -C_\varepsilon(S,\pi) \qquad \text{pour } \varepsilon = 0$$

Finalement par la propriété (2.9),

$$q'(\pi, \mu(S)) \leq -C_\varepsilon(S, \pi) \qquad \forall \varepsilon \geq 0. \square$$

Un corollaire immédiat est le suivant :

$\forall\ \varepsilon \geq 0, \quad C_\varepsilon(S, \pi) > 0 \ \Rightarrow\ \mu(S)$ est une direction de descente de la fonction duale q le long de π.

Ainsi, ces deux résultats servent à caractériser la direction de descente duale à partir des bornes-ε. Rappelons que dans le cas des coûts linéaires, la dernière étape de l'algorithme consistait en une maximisation de la fonction duale et par conséquent engendrait la recherche d'une direction de montée dans la méthode. Tandis que pour le cas des coûts convexes, nous avons converti l'optimisation du problème dual en une minimisation, d'où la recherche d'une direction de descente.

Soit (x, π) une paire primale-duale de vecteurs satisfaisant les (ε-CC'). Une séquence de noeuds $\{n_1, n_2, \cdots, n_k\}$ est une *chaîne d'augmentation de flot,* que nous dénotons ε-*CAF* pour la différencier de celle définie dans le cas des coûts linéaires, si $d_{n_1} < 0, \quad d_{n_k} > 0$ et pour $m = 1, 2, \cdots, k-1$, ou bien il existe un arc (n_{n_m}, n_{m+1}) avec $x_{n_m n_{m+1}} < u^{\varepsilon}_{n_m n_{m+1}}$, ou bien un arc (n_{m+1}, n_m) avec $x_{n_{m+1} n_m} > l^{\varepsilon}_{n_{m+1} n_m}$.

Une solution réalisable (x, π) est ε-*optimale*, si $f(x) + q(\pi)$ diffère de l'optimum ($f(x^*) + q(\pi^*) = 0$) d'au plus ε.

Nous présentons maintenant l'algorithme de relaxation primal-dual pour les coûts convexes.

II.2.4 Algorithme

L'algorithme, paramétrisé par un scalaire positif ε, est basé sur l'idée des ε–conditions de complémentarité. Chaque itération commence avec un couple primal-dual de vecteurs satisfaisant les (ε–CC'). Si tous les déficits aux noeuds sont nuls, l'algorithme s'arrête. Sinon, on utilise la règle d'étiquetage de Ford et Fulkerson (1961) jusqu'à ce qu'on trouve une chaîne d'augmentation de flot ou une direction duale de descente. Dans le premier cas, on modifie le flot sur la chaîne afin de diminuer les déficits, c'est-à-dire réduire la non-réalisabilité primale. Dans le second cas, on effectue une minimisation pour déterminer le pas à faire dans la direction trouvée. Le processus est itératif : il continue jusqu'à ce que tous les déficits deviennent nuls.

Nous présentons dans les lignes qui suivent une itération de l'algorithme qui opère sur un noeud positif. Une itération construite à partir d'un noeud avec un déficit négatif est fournie à l'appendice B.

Itération de l'algorithme de relaxation (coûts convexes)

Etape 0. Soient x et π satisfaisant les (ε–CC').

Etape 1. Si $d_i = 0 \ \forall i \in N$ alors *STOP*.
Sinon faire : $M = \emptyset$
$S = \emptyset$

où

M = ensemble des sommets marqués non-examinés

S = ensemble des sommets marqués et examinés.

Choisir un sommet i tel que $d_i > 0$. Poser $S = \emptyset$.

Aller à l'étape 2.

Etape 2. **Choisir un sommet $k \in M$.**

Faire : $S \leftarrow S \cup \{k\}$

$M \leftarrow M - \{k\}$.

Aller à l'étape 3.

Etape 3. **Examiner le sommet k en marquant tous les sommets non-marqués $j \in \Gamma(k)$ et en leur donnant l'étiquette k de la manière suivante :**

$$M \leftarrow M \cup \{j\} \quad \text{si} \quad \begin{cases} (k, j) \in A \mid x_{kj} > l_{kj}^{\varepsilon} \\ \text{ou} \\ (j, k) \in A \mid x_{jk} < u_{jk}^{\varepsilon}. \end{cases}$$

Si $C_{\varepsilon}(S, \pi) > 0$ aller à l'étape 5.

Sinon,

s'il existe un sommet $j \in M$ étiqueté à partir de k *t.q.* $d_j < 0$ aller à l'étape 4.

Sinon retourner à l'étape 2.

Etape 4. Retracer la ε–*CAF* reliant les sommets j et i (j est le sommet marqué avec $d_j < 0$). Soit δ la capacité de la chaîne. Augmenter de δ le flot sur les arcs orientés de j vers i et enlever δ unités de flot sur les arcs orientés de i vers j. Ajuster les déficits aux extrémités de la chaîne. Passer à l'itération suivante.

Etape 5. **Déteminer λ^* *t.q.***

$$q(\pi + \lambda^* \mu(S)) = \min_{\lambda > 0} \{q(\pi + \lambda \mu(S)\}.$$

Faire $\pi \leftarrow \pi + \lambda^* \mu(S)$. **Mettre à jour des bornes–ε ainsi que le vecteur de flot afin de satisfaire les (ε–*CC'*).**

Passer à l'itération suivante.

II.2.5 Convergence de l'algorithme

Bertsekas, Hossein et Tseng (1987) démontrent, sous certaines hypothèses relatives à la programmation monotropique, que l'algorithme de relaxation se termine en un nombre fini d'itérations avec une solution ε–optimale. Etant donné que nous avons adopté une approche différente à celle suivie par ces auteurs, nous avons eu recours à de nouvelles hypothèses. De ce fait, afin d'utiliser le même raisonnement qu'eux, nous devons démontrer que nos hypothèses constituent une condition suffisante pour satisfaire les leurs. Ceci fait l'objet de la proposition suivante :

Proposition 2.3

Soient les deux hypothèses suivantes :

I) chacune des fonctions f_{ij} est convexe, continue et il existe une solution réalisable pour $(P\,2)$

II) $(P\,2)$ est stable

alors I et II constituent une condition suffisante pour satisfaire les hypothèses suivantes :

I′) chaque f_{ij} est convexe, semi-continue inférieurement et il existe une solution réalisable pour $(P\,2)$

II′) la fonction $g_{ij}(t_{ij})$ définie en (2.6) possède une solution réelle $\forall\, t_{ij} \in R$

III′) $f_{ij}'(x_{ij})^{+} > -\infty$ et $f_{ij}'(x_{ij})^{-} < +\infty \qquad \forall\, x_{ij}$.

preuve:

Il est clair qu'une fonction continue est semi-continue inférieurement et remplit la condition III′. Ainsi, les conditions I′ et III′ sont satisfaites. D'autre part, étant donné que le problème primal est stable, le problème dual possède une solution optimale (cf. appendice A), satisfaisant ainsi l'hypothèse II′. Par

conséquent, les hypothèses I et II sont une condition suffisante aux hypothèses I′, II′. □

Ainsi nous pouvons nous servir des arguments utilisés par Bertsekas, Hossein et Tseng (1987) pour démontrer la convergence de l'algorithme de relaxation vers une solution ε–optimale. Pour les détails de la preuve, nous référons le lecteur à la dernière référence citée.

La preuve se fait en montrant d'abord que chacune des étapes d'une itération de l'algorithme est bien définie et qu'une itération se termine en un nombre fini d'opérations. Ensuite, ces auteurs montrent que l'algorithme de relaxation, lorsqu'appliqué conjointement avec une règle d'étiquetage facile à implanter, se termine en un nombre fini d'étapes. La preuve se fait en deux séquences.

Il s'agit d'une part, de montrer que le nombre de pas duaux n'est pas infini. L'argumentation vient du fait que le coût dual est nécessairement $-\infty$ si le nombre de pas duaux est infini. D'autre part, on doit montrer que le nombre de chaînes d'augmentation de flot entre des pas duaux successifs est fini. Cela se fait en montrant que le nombre de chaînes d'augmentation est fini sous une règle d'étiquetage appropriée pour l'algorithme de relaxation. Par ailleurs, puisque ce dernier se termine seulement lorsque tous les noeuds ont un déficit nul, alors le vecteur flot final est primal-réalisable.

De plus, puisque les (ε–CC') sont maintenues satisfaites à chacune des itérations de l'algorithme, il s'ensuit que la paire finale (x, π) satisfait également les (ε–CC'). On peut alors démontrer, comme il est fait dans Bertsekas, Hossein et Tseng (1987) que la solution optimale générée par l'algorithme peut être amenée aussi près que désiré de la solution optimale réelle, en prenant ε suffisamment petit. La partie essentielle de l'argumentation est résumée par la proposition suivante dont la preuve se trouve dans le même article.

Proposition 2.3

Soient $x(\varepsilon)$ et $\pi(\varepsilon)$ la paire de vecteurs flot et prix tels que $(x(\varepsilon), \pi(\varepsilon))$ satisfont les (ε–CC') et $x(\varepsilon)$ satifait les équations de conservation de flot, alors

$$f(x(\varepsilon)) + q(\pi(\varepsilon)) \rightarrow 0 \quad \text{lorsque} \quad \varepsilon \rightarrow 0.$$

Toutefois, cette proposition ne dit pas quelle valeur doir prendre ε pour atteindre un certain degré d'optimalité. Il s'agit donc à priori de résoudre le problème pour une valeur de ε afin d'obtenir $x(\varepsilon)$ et $\pi(\varepsilon)$, puis d'évaluer la qualité de la solution en calculant l'*écart dual*† et décider alors si ε doit être diminué.

En ce qui concerne les expérimentations numériques, dans le cas linéaire, l'algorithme de relaxation, dont le code porte le nom *RELAX-II*, s'est révélé très performant. En effet, les résultats démontrent que *RELAX-II* est beaucoup plus efficace que deux autres programmes informatiques bien connus, soient *KILTER* (Aashtiani et Magnanti, 1976), un algorithme de type primal-dual et *RNET* (Grigoriadis et Hsu, 1980) de type primal. Ces résultats encourageants ont été une motivation en faveur du choix des méthodes de relaxation.

Pour les coûts convexes cependant, nous trouvons très peu de résultats dans la littérature. Leur code, dénommé *MNRELAX*, n'a été testé que sur des problèmes quadratiques. De toute façon, nous ne tenons pas à tester la performance de l'algorithme, mais nous désirons plutôt un algorithme qui résout notre problème. D'où l'attrait de *MNRELAX* qui solutionne le problème mixte linéaire et convexe.

Nous possédons maintenant tous les outils en main pour présenter notre méthode de résolution pour le problème de localisation-distribution avec échanges interdépôts. C'est le sujet du troisième chapitre.

---

† Ce vocabulaire découle de la terminologie anglaise "duality gap".

CHAPITRE III

APPROCHE DE RESTRICTION

Ce chapitre constitue le coeur de ce mémoire, il est dédié à l'approche que nous proposons pour aborder le problème de localisation-distribution avec échanges interdépôts énoncé au chapitre I. Nous dérivons deux nouvelles formulations mathématiques du modèle et développons un algorithme de résolution. Il s'agit de notre contribution apportée à l'étude du problème particulier de localisation auquel font face les grandes compagnies de transport maritime et terrestre de conteneurs.

Rappelons que ce problème, tel que formulé originellement, peut atteindre rapidement une très grande taille, ce qui le rend difficile à traiter. Ainsi, nous n'attaquons pas directement (L), mais nous étudions plutôt la relaxation forte de ce problème qui possède, comme nous l'avons vu, une structure de réseau. Remarquons qu'il suffit d'une simple substitution de variables pour éliminer de la formulation de (SRL) les contraintes de type (1.12) et (1.13) qui rendent le problème difficile à résoudre directement (voir, à cet effet, Schrage, 1975, 1978). Or, cette transformation nous permet de déduire un problème équivalent à (SRL), dans lequel apparaît également une structure de réseau et qui, en plus, a l'avantage de ne contenir que des variables de flot. Cependant une petite difficulté réside : le problème n'est pas séparable. Par conséquent, nous ne pouvons pas appliquer directement l'algorithme

de relaxation primal-dual comme technique de résolution. Nous surmontons cette embûche en dérivant un sous-problème au problème transformé, défini sur une portion seulement du domaine réalisable initial. Ce dernier peut alors être résolu par *MNRELAX*, auquel nous apportons d'importantes modifications requises par la nature particulière de notre problème. Notre approche est une restriction dans le sens où nous restreignons le domaine réalisable du problème. La résolution d'un sous-problème génère donc une borne supérieure sur la valeur optimale de la fonction économique du problème initial.

Ainsi, nous proposons une approche de restriction qui consiste à générer plusieurs sous-problèmes définis sur différentes portions du domaine réalisable, portions que nous pouvons modifier localement ou globalement. Ceci nous amène à développer deux versions algorithmiques, toutes deux présentées dans ce chapitre. La résolution de chacun des sous-problèmes se fait en adaptant l'algorithme pour les coûts convexes décrit au chapitre précédent.

La progression de ce chapitre de déroulera comme suit. Avant tout, nous discutons des deux approches de résolution envisagées. Selon l'approche choisie, nous reformulons le problème, énonçons le modèle résultant de l'élimination des variables d'ouverture des dépôts dans la relaxation forte et analysons ses propriétés. Nous formulons ensuite le sous-problème obtenu par le procédé de restriction et donnons ses propriétés. Puis, nous introduisons le vocabulaire nécessaire pour présenter clairement notre méthode. Nous énonçons ensuite le problème dual du sous-problème et donnons ses propriétés. Nous ajoutons de nouvelles définitions et notations qui découlent du problème dual. Nous terminons ce chapitre par la présentation de notre algorithme, incluant les deux versions mentionnées auparavant. Nous l'exposons, dans un premier temps, de façon synthétique tout en soulignant ses particularités. Puis, dans un second temps, nous explicitons formellement ses particularités, en plus des modifications que nous apportons à *MNRELAX*.

III.1 Approches de résolution

Une première tentative de résolution vise à appliquer l'algorithme de relaxation primal-dual à la relaxation forte de (*PRL*) énoncée au chapitre I. Rappelons que ce problème est obtenu en appliquant la méthode des pénalités au problème de localisation-distribution avec contraintes d'équilibrage. Récrivons la fonction économique de (*PRL*) :

$$z = \sum_{j \in D} f_j y_j + \sum_{p \in P} \sum_{j \in D} \left[\sum_{i \in C} (c_{ijp} x_{ijp} + c_{jip} x_{jip}) + \sum_{k \in D} s_{kjp} w_{kjp} \right]$$

$$+ \sum_{p \in P} \sum_{i \in C} \sum_{j \in D} \left[K_{ijp} (\min\{0, O_{ip} y_j - x_{ijp}\})^2 + K_{jip} (\min\{0, D_{ip} y_j - x_{jip}\})^2 \right] \qquad (3.1)$$

A première vue, la nature convexe de l'objectif, en plus de la structure de réseau que possède la relaxation forte de (*PRL*), nous porte à croire qu'il est possible de résoudre cette dernière en appliquant un algorithme efficace de flot à coût convexe, en particulier *MNRELAX*. Cependant, un examen plus approfondi de la fonction définie par (3.1) révèle qu'elle n'est pas séparable. En effet, les termes quadratiques de l'objectif dissimulent des termes mixtes dans lesquels apparaissent deux sortes de variables : les variables d'ouverture des dépôts, ainsi que les variables de flot x_{ijp} et x_{jip}. Néanmoins, la recherche nécessaire à l'étude d'un tel problème dépasse le cadre de ce travail.

Nous avons donc trouvé une alternative de résolution qui nous semble intéressante. Elle porte sur la relaxation forte du problème de localisation-distribution avec échanges interdépôts. Tout d'abord reformulons la fonction objectif de (*SRL*) ainsi que les contraintes qui compliquent la résolution de ce problème :

$$z = \sum_{j \in D} f_j y_j + \sum_{p \in P} \sum_{j \in D} \left[\sum_{i \in C} (c_{ijp} x_{ijp} + c_{jip} x_{jip}) + \sum_{k \in D} s_{kjp} w_{kjp} \right]$$

$$x_{ijp} \leq O_{ip} y_j \qquad \forall\, i \in C,\ j \in D,\ p \in P \qquad (3.2)$$

$$x_{jip} \leq D_{ip} y_j \qquad \forall\, i \in C,\ j \in D,\ p \in P \qquad (3.3)$$

Outre les contraintes de non-négativité, les variables d'ouverture des dépôts y_j n'interviennent que dans les contraintes de type (3.2) et (3.3). Ainsi, à l'optimum, par l'agrégation de ces contraintes, il est possible d'obtenir une expression pour chacune des variables d'ouverture des dépôts. Cette dernière correspond au maximum des ratios flot/offre et flot/demande pour un dépôt j donné. L'objectif peut alors se récrire en substituant les y_j par l'expression obtenue pour ainsi les éliminer de la formulation de (*SRL*). Le nouveau problème contient alors seulement des variables de flot, sans toutefois être séparable. Cette dernière difficulté, de moindre mal, reste la seule à laquelle nous faisons face. Nous la surmontons en adoptant une approche de restriction qui nous permet d'adapter la méthode de relaxation primale-duale comme technique de résolution.

Nous présentons, dans les deux prochaines sections, les développements analytiques de deux modèles mathématiques résultant respectivement de la transformation de (*SRL*) et de la restriction.

III.2 Problème transformé

Dans cette section, nous reformulons la relaxation forte du problème de localisation-distribution avec contraintes d'équilibrage, énonçons le modèle résultant de l'élimination des variables d'ouverture des dépôts, et analysons les propriétés du problème transformé.

Ecrivons à nouveau la relaxation forte de (L) :

$$(SRL)\quad \textit{Minimiser}\quad z = \sum_{j\in D} f_j y_j + \sum_{p\in P}\sum_{j\in D}\left[\sum_{i\in C_1} c_{ijp}x_{ijp} + \sum_{i\in C_2} c_{jip}x_{jip} + \sum_{k\in D} s_{kjp}w_{kjp}\right]$$

sujet à

$$\sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C_1,\ p\in P$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C_2,\ p\in P$$

$$x_{ijp} \le O_{ip}\, y_j \qquad \forall\, i\in C_1,\ p\in P \qquad (3.2)$$

$$x_{jip} \le D_{ip}\, y_j \qquad \forall\, i\in C_2,\ p\in P \qquad (3.3)$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D,\ p\in P$$

$$x_{ijp} \ge 0 \qquad \forall\, i\in C_1,\ j\in D,\ p\in P$$

$$x_{jip} \ge 0 \qquad \forall\, i\in C_2,\ j\in D,\ p\in P$$

$$w_{kjp} \ge 0 \qquad \forall\, k,\ j\in D,\ p\in P$$

$$y_j \ge 0 \qquad \forall\, j\in D$$

où

$C_1 = \{\ i\in C \mid \text{client-origine}\ \}$
$C_2 = \{\ i\in C \mid \text{client-destination}\ \}.$

Notons qu'en pratique $C_1 = C_2 = C$.

Soit (x^*, y^*) une solution optimale de ce problème. Les contraintes de type (3.2) et (3.3) impliquent alors que :

$$y_j^* = \max \left\{ \max_{\substack{i \in C_1 \\ p \in P}} \left\{ \frac{x_{ijp}^*}{O_{ip}} \right\}, \max_{\substack{i \in C_2 \\ p \in P}} \left\{ \frac{x_{jip}^*}{D_{ip}} \right\} \right\} \qquad \forall j \in D \qquad (3.4)$$

Ce résultat réunit toutes les contraintes où interviennent les y_j, incluant les contraintes de non-négativité sur les variables d'ouverture des dépôts. En effet, ces dernières deviennent alors redondantes étant donné la restriction de non-négativité sur les variables de flot. En substituant, dans (*SRL*), chacune des variables d'ouverture des dépôts par l'expression définie en (3.4), nous les éliminons de la formulation du modèle afin de générer un problème équivalent dans lequel n'apparaissent plus que des variables de flot. Soit (*SRLT*), le problème transformé. Il s'écrit ainsi :

$$(SRLT) \quad \textit{Minimiser} \quad z = \sum_{j \in D} f_j \max \left\{ \max_{\substack{i \in C_1 \\ p \in P}} \left\{ \frac{x_{ijp}}{O_{ip}} \right\}, \max_{\substack{i \in C_2 \\ p \in P}} \left\{ \frac{x_{jip}}{D_{ip}} \right\} \right\}$$

$$+ \sum_{p \in P} \sum_{j \in D} \left\{ \sum_{i \in C_1} c_{ijp} x_{ijp} + \sum_{i \in C_2} c_{jip} x_{jip} + \sum_{k \in D} s_{kjp} w_{kjp} \right\}$$

sujet à

$$\sum_{j \in D} x_{ijp} = O_{ip} \qquad \forall i \in C_1, p \in P$$

$$\sum_{j \in D} x_{ijp} = D_{ip} \qquad \forall i \in C_2, p \in P$$

$$\sum_{i \in C} x_{ijp} + \sum_{k \in D} w_{kjp} - \sum_{i \in C} x_{jip} - \sum_{k \in D} w_{jkp} = 0 \qquad \forall j \in D, p \in P$$

$$x_{ijp} \geq 0 \qquad \forall i \in C_1, j \in D, p \in P$$

$$x_{jip} \geq 0 \qquad \forall i \in C_2, j \in D, p \in P$$

$$w_{kjp} \geq 0 \qquad \forall k, j \in D, p \in P.$$

(*SRLT*) possède la propriété suivante :

Proposition 3.1

(*SRLT*) est un problème de multiflot à coût minimum sans capacités à coûts convexes.

preuve:

Notons d'abord que les domaines réalisables de la relaxation forte de (*L*) et de (*SRLT*) sont identiques. Ainsi, puisque le problème "relaxé" possède la structure de multiflot à coût minimum (Crainic, Dejax et Delorme, 1986), il en est de même pour (*SRLT*). De plus, les fonctions de coût sont linéaires pour tous les arcs interdépôts. Alors que pour les arcs dépôts-clients (clients-dépôts), les fonctions de coût sont linéaires par morceaux avec deux morceaux, dont la pente du premier morceau est c_{ijp} (c_{jip}) et celle du deuxième morceau est $c_{ijp} + \frac{f_j}{O_{ip}}$ ($c_{jip} + \frac{f_j}{D_{ip}}$). Ainsi, puisque la pente du deuxième morceau linéaire est supérieure à celle du premier morceau, selon l'hypothèse de non-négativité des coûts unitaires, ces fonctions sont convexes. La fonction objectif, composée de termes simplement linéaires et linéaires par morceaux de pentes non-négatives croissantes, est donc convexe. □

D'autre part, (*SRLT*) est un problème non-séparable en x et w. Cette propriété provient du fait que la fonction de coût associée à un arc client-dépôt $i'jp'$ (dépôt-client $ji'p'$), linéaire en deux morceaux, possède un morceau qui dépend du flot sur tous les autres arcs clients-dépôts ijp et dépôts-clients jip connectés au même dépôt j. Pour éliminer le caractère non-séparable que possède le problème, nous adoptons une approche de restriction de manière à résoudre le problème sur une partie de son domaine réalisable, partie pouvant être modifiée tant que l'optimum n'est pas atteint. En fait, nous dérivons de (*SRLT*), un sous-problème qui se distingue par des capacités imposées sur certaines variables; d'où le sens d'une approche de restriction. Nous consacrons le reste de ce chapitre à la présentation de l'approche proposée.

III.3 Sous-problème restreint et propriétés

Avant de formaliser le sous-problème, introduisons certaines notations. Nous définissons pour chacun des dépôts un *coefficient d'ouverture* dénoté $\overline{y}_j$, ainsi qu'une *variable restrictive* associée à un des arcs clients-dépôts et dépôts-clients relatifs au dépôt considéré. Tous les autres arcs clients-dépôts et dépôts-clients reliés au même dépôt sont appelés *arcs restreints*. Le sous-problème restreint se distingue alors de (*SRLT*) par des capacités, définies à partir des coefficients d'ouverture des dépôts, que nous imposons sur les variables restreintes. D'autre part, la variable restrictive ne possède pas de contrainte explicite de capacité. Cette dernière est plutôt remplacée par un coût à payer qui la pénalise lorsqu'elle est violée. Cette pénalité correspond au coût d'ouverture du dépôt auquel est associée la variable restrictive.

Formulons maintenant le sous-problème (*SRLT*$(\overline{y})$), appelé *sous-problème restreint*, à résoudre en fonction du vecteur (donné) des coefficients d'ouverture des dépôts $\overline{y}$:

$$(SRLT(\overline{y})) \quad \textit{Minimiser} \quad z = \sum_{j \in D} \left[f_j \max \left\{ \overline{y}_j \, , \, \frac{x_j^R}{a_j^R} \right\} + c_j^R x_j^R \right]$$

$$+ \sum_{j \in D} \sum_{ijp \in R_1(j)} c_{ijp} x_{ijp} + \sum_{j \in D} \sum_{jip \in R_2(j)} c_{jip} x_{jip}$$

$$+ \sum_{j \in D} \sum_{p \in P} \sum_{k \in D} s_{kjp} w_{kjp}$$

sujet à

$$\sum_{j \in D} x_{ijp} = O_{ip} \qquad \forall\, i \in C_1, p \in P$$

$$\sum_{j \in D} x_{jip} = D_{ip} \qquad \forall\, i \in C_2, p \in P$$

$$x_{ijp} \leq O_{ip}\ \bar{y}_j \qquad \forall\, ijp \in R_1(j),\ j \in D \qquad (3.5)$$

$$x_{jip} \leq D_{ip}\ \bar{y}_j \qquad \forall\, jip \in R_2(j),\ j \in D \qquad (3.6)$$

$$\sum_{i \in C} x_{ijp} + \sum_{k \in D} w_{kjp} - \sum_{i \in C} x_{jip} - \sum_{k \in D} w_{jkp} = 0 \qquad \forall\, j \in D,\ p \in P$$

$$x_{ijp} \geq 0 \qquad \forall\, i \in C_1,\ j \in D,\ p \in P$$

$$x_{jip} \geq 0 \qquad \forall\, i \in C_2,\ j \in D,\ p \in P$$

$$w_{kjp} \geq 0 \qquad \forall\, k,\ j \in D,\ p \in P$$

où pour tout $j \in D$:

x_j^R : la variable dite restrictive associée au dépôt j; elle est choisie parmi une des variables x_{ijp}, $i \in C_1$, $p \in P$, x_{jip}, $i \in C_2$, $p \in P$

$$a_j^R \equiv \begin{cases} O_{ip} & \text{si } ijp \text{ définit } x_j^R,\ i, p \text{ fixés} \\ D_{ip} & \text{si } jip \text{ définit } x_j^R,\ i, p \text{ fixés} \end{cases}$$

$$c_j^R \equiv \begin{cases} c_{ijp} & \text{si } ijp \text{ définit } x_j^R,\ i, p \text{ fixés} \\ c_{jip} & \text{si } jip \text{ définit } x_j^R,\ i, p \text{ fixés} \end{cases}$$

$$R_1(j) = \{ijp \mid x_{ijp} \neq x_j^R,\quad \forall\, i \in C_1, p \in P\}$$
$$R_2(j) = \{jip \mid x_{jip} \neq x_j^R,\quad \forall\, i \in C_2, p \in P\}.$$

$R(j) = R_1(j) \cup R_2(j)$ est l'ensemble des arcs restreints reliés au dépôt j. En d'autres mots, $R(j)$ contient tous les arcs de la forme client-dépôt et dépôt-client associés au dépôt j sauf l'arc restrictif, lui-même étant un arc de la forme client-dépôt ou de la forme dépôt-client, relié à ce dépôt.

Soulignons que le sous-problème restreint, étant déterminé par les coefficients d'ouverture des dépôts, nous dirons que les $\bar{y}_j$ définissent le *domaine réalisable*

restreint **du sous-problème en question.**

Remarquons que le problème ne possède pas de capacités explicites sur le flot des arcs interdépôts. Cependant, nous déterminons, pour les fins d'utilisation de l'algorithme de relaxation, une borne supérieure sur le flot pouvant circuler sur ces arcs. Pour l'instant, nous la posons égal à u_{kjp}.

($SRLT(\bar{y})$) possède la propriété qui suit :

Proposition 3.2

($SRLT(\bar{y})$) est un problème de multiflot à coût minimum à coûts convexes avec capacité sur les variables restreintes.

preuve:

Notons d'abord que le domaine réalisable du problème ($SRLT(\bar{y})$) ne diffère de celui du problème ($SRLT$) que par les contraintes (3.5) et (3.6). Ces dernières correspondent aux contraintes de capacité sur les variables retreintes. Ainsi, par la proposition (3.1), ($SRLT(\bar{y})$) possède la structure d'un problème de flot à coût minimum avec capacité. De plus, les fonctions de coût sont linéaires pour tous les arcs interdépôts ainsi que pour tous les arcs restreints (cf. figures 3.1 et 3.2). Tandis que, pour les arcs restrictifs, les fonctions de coût sont linéaires par morceaux avec deux morceaux dont la pente du premier morceau est supérieure à celle du second morceau, selon l'hypothèse de non-négativité des coûts unitaires (cf. figure 3.3). La fonction objectif est par conséquent convexe; ce qui complète la preuve de cette proposition. □

Aussi, ($SRLT(\bar{y})$) est clairement un problème séparable puisque chaque fonction de coût associé à un arc dépend uniquement du flot sur cet arc. On peut donc utiliser *MNRELAX* pour résoudre ce problème, en y apportant certaines modifications nécessitées par la nature particulière du sous-problème à résoudre.

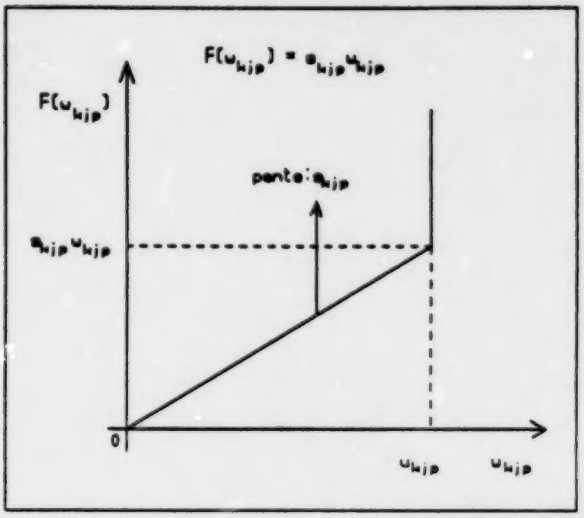


Figure 3.1. Graphe de la fonction de coût associée à un arc interdépôt pour un produit p donné.

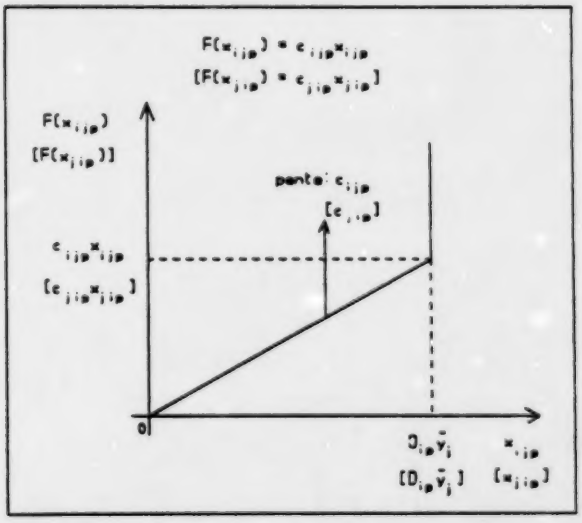


Figure 3.2. Graphe de la fonction de coût associée à un arc restreint du dépôt j pour un produit p donné.

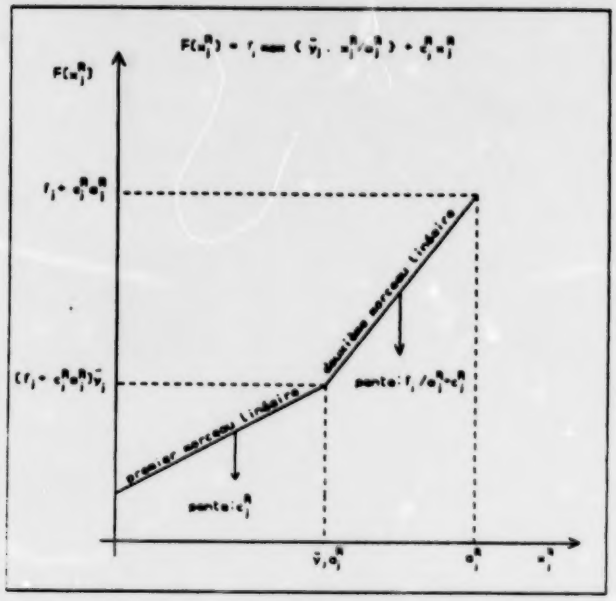


Figure 3.3. Graphe de la fonction de coût associée à l'arc restrictif du dépôt j.

III.4 Définitions et notations

Dans cette section, nous introduisons les définitions et les notations relatives au problème primal, qui serviront à la présentation de l'algorithme de restriction.

Soit $\bar{x}$, une valeur connue du vecteur x. Nous supposons que la variable restrictive associée à un dépôt j donné est telle que :

$$\frac{\bar{x}_j^R}{a_j^R} \geq \begin{cases} \dfrac{\bar{x}_{ijp}}{O_{ip}} & \forall\, ijp \in R_1(j) \\ \dfrac{\bar{x}_{jip}}{D_{ip}} & \forall\, jip \in R_2(j). \end{cases} \tag{3.7}$$

Définissons :

$$\frac{\bar{x}_j^m}{a_j^m} = \max \left\{ \max_{ijp \in R_1(j)} \left\{ \frac{\bar{x}_{ijp}}{O_{ip}} \right\}, \max_{jip \in R_2(j)} \left\{ \frac{\bar{x}_{jip}}{D_{ip}} \right\} \right\} \quad \forall\, j \in D \qquad (3.8)$$

où a_j^m est défini comme a_j^R. Si plus d'un arc satisfait la relation (3.8), alors l'arc est choisi de façon arbitraire. (3.8) définit, pour chacun des dépôts, le maximum des rapports flot/offre et flot/demande parmi toutes les variables restreintes relatives au dépôt considéré.

Des relations (3.7) et (3.8), on peut déduire la relation suivante :

$$\frac{\bar{x}_j^R}{a_j^R} \geq \frac{\bar{x}_j^m}{a_j^m} \quad \forall\, j \in D. \qquad (3.9)$$

De plus, nous supposons que le coefficient d'ouverture du dépôt j est tel que :

$$\bar{y}_j \in \left[\frac{\bar{x}_j^m}{a_j^m}, \frac{\bar{x}_j^R}{a_j^R} \right] \quad \forall\, j \in D. \qquad (3.10)$$

En plus de déterminer les bornes supérieures sur les variables restreintes, les coefficients d'ouverture des dépôts servent également à calculer les bornes intermédiaires (supérieures ou inférieures) sur les variables restrictives, selon le morceau linéaire de la fonction où on se situe (premier morceau ou deuxième morceau) (cf. figure 3.3). Les coefficients d'ouverture des dépôts représentent dans quelle proportion le dépôt est ouvert; $\bar{y}_j \in [0,1]$.

Ainsi, par définition,

$$\bar{y}_j \geq \frac{\bar{x}_j^m}{a_j^m} \qquad \forall\, j \in D.$$

Finalement, en se rapportant à la méthode de relaxation primale-duale, nous définissons, pour un produit p donné, le déficit au noeud n, d_n par :

$$d_i = \begin{cases} \sum_{j \in D} x_{ijp} - O_{ip} & \forall\, i \in C_1 \\ -\sum_{j \in D} x_{jip} + D_{ip} & \forall\, i \in C_2 \end{cases} \qquad (3.11)$$

$$d_j = \sum_{i \in C_1} x_{jip} + \sum_{k \in D} w_{jkp} - \sum_{i \in C_2} x_{jip} - \sum_{k \in D} w_{kjp} \quad \forall\, j \in D.$$

III.5 Problème dual du sous-problème restreint

Ecrivons le problème dual de $(SRLT(\overline{y}))$:

$$(DT) \qquad \underset{\pi}{Minimiser}\ \Theta = \sum_{\substack{\alpha \in A \\ \alpha=(m,\,n)}} \sup_{v_\alpha \in \Omega_\alpha} \left\{ -F_\alpha(v_\alpha) + (\pi_m - \pi_n) v_\alpha \right\}$$

$$- \sum_{p \in P} \sum_{i \in C_1} \pi_{ip} O_{ip} + \sum_{p \in P} \sum_{i \in C_2} \pi_{ip} D_{ip}$$

où

α : un arc du réseau reliant les noeuds m et n; $m, n \in N$

v_α : le flot sur l'arc α

Ω_α : l'intervalle de variation du flot sur l'arc α

$$\Omega_\alpha = [\, l_\alpha , u_\alpha \,] = \begin{cases} [0 , a_j^R] & \text{si } \alpha \text{ est un arc restrictif} \\ [0 , O_{ip}\ \overline{y}_j] & \text{si } \alpha \equiv ijp \in R_1(j),\ j \in D \\ [0 , D_{ip}\ \overline{y}_j] & \text{si } \alpha \equiv jip \in R_2(j),\ j \in D \\ [0 , u_{kjp}] & \text{si } \alpha \equiv kjp,\ k, j \in D,\ p \in P \end{cases}$$

$F_\alpha(v_\alpha)$: la fonction de coût associée à l'arc α

$$F_\alpha(v_\alpha) = \begin{cases} f_j \max\left\{\overline{y}_j\ ,\ \dfrac{v_\alpha}{a_j^R}\right\} + c_j^R v_\alpha & \text{si } \alpha \text{ est un arc restrictif} \\ c_{ijp}\, v_\alpha & \text{si } \alpha \equiv ijp \in R_1(j),\ j \in D \\ c_{jip}\, v_\alpha & \text{si } \alpha \equiv jip \in R_2(j),\ j \in D \\ s_{kjp}\, v_\alpha & \text{si } \alpha \equiv kjp,\ k,\ j \in D,\ p \in P \end{cases}$$

π_m : le prix au noeud m

$$\pi_m = \begin{cases} \pi_{ip} & \text{si } m = ip,\ i \in C_1 \cup C_2,\ p \in P \\ \pi_{jp} & \text{si } m = jp,\ j \in D,\ p \in P. \end{cases}$$

De plus, posons :

$$g_\alpha(t_\alpha) = \sup_{v_\alpha \in \Omega_\alpha} \left\{ -F_\alpha(v_\alpha) + t_\alpha v_\alpha \right\} \qquad \forall\, \alpha \in A.$$

où t_α correspond à la tension associée à l'arc α (cf. section II.2.3). g_α est la fonction de coût duale associée à l'arc α. Les graphes des fonctions duales pour les trois types d'arcs de notre problème sont tracés dans les figures 3.4 à 3.6.

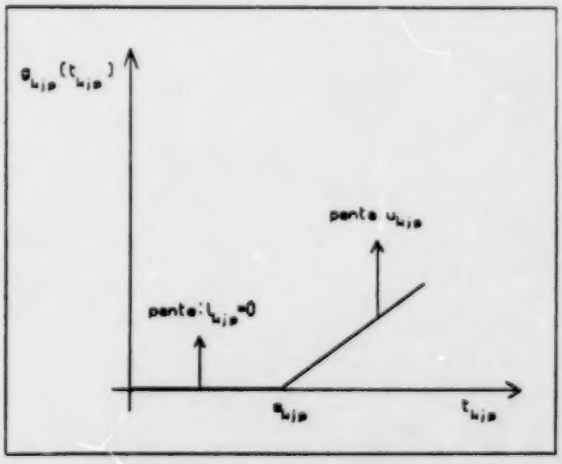


Figure 3.4. Graphe de la fonction de coût duale associée à un arc interdépôt pour un produit p donné.

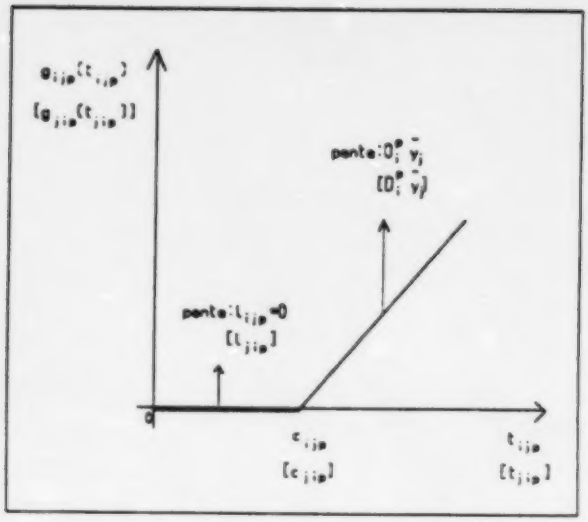


Figure 3.5. Graphe de la fonction de coût duale associée à un arc restreint d'un dépôt j pour un produit p donné.

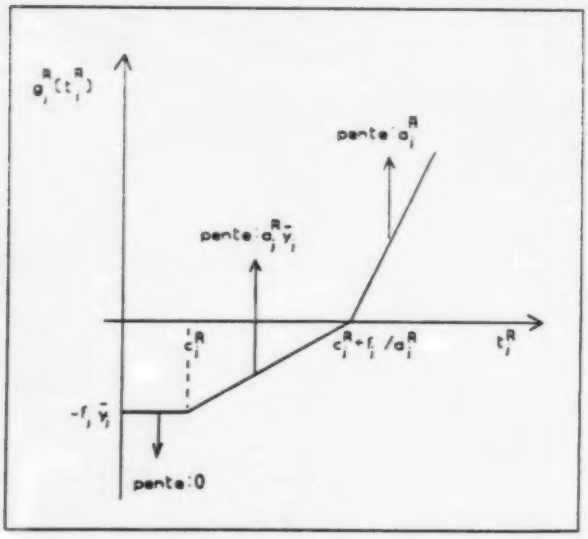


Figure 3.6. Graphe de la fonction de coût duale associée à l'arc restrictif d'un dépôt j.

III.6 Autres définitions et notations

Suite au problème dual, nous ajoutons d'autres définitions et notations à celles déjà introduites.

Se référant à la terminologie introduite dans l'article de Bertsekas (1985) pour les coûts linéaires, nous extrapolons les définitions utilisées pour dire que l'arc restrictif relatif au dépôt j est :

$$
\begin{aligned}
&\varepsilon\text{–}inactif^- && \text{si} \quad t_j^R < c_j^R - \varepsilon, \\
&\varepsilon\text{–}libre^- && \text{si} \quad c_j^R - \varepsilon \le t_j^R \le c_j^R + \varepsilon, \\
&\varepsilon\text{–}actif^- && \text{si} \quad c_j^R + \varepsilon < t_j^R < c_j^R + \frac{f_j}{a_j^R} - \varepsilon, \\
&\varepsilon\text{–}inactif^+ && \text{si} \quad c_j^R + \varepsilon < t_j^R < c_j^R + \frac{f_j}{a_j^R} - \varepsilon, \\
&\varepsilon\text{–}libre^+ && \text{si} \quad c_j^R + \frac{f_j}{a_j^R} - \varepsilon \le t_j^R \le c_j^R + \frac{f_j}{a_j^R} + \varepsilon, \\
&\varepsilon\text{–}actif^+ && \text{si} \quad t_j^R > c_j^R + \frac{f_j}{a_j^R} + \varepsilon,
\end{aligned}
$$

et qu'un arc restreint ou interdépôt α est :

$$
\begin{aligned}
&\varepsilon\text{–}inactif && \text{si} \quad t_\alpha > c_\alpha - \varepsilon, \\
&\varepsilon\text{–}libre && \text{si} \quad c_\alpha - \varepsilon \le t_\alpha \le c_\alpha + \varepsilon \quad \text{et} \\
&\varepsilon\text{–}actif && \text{si} \quad t_\alpha > c_\alpha + \varepsilon.
\end{aligned}
$$

Les graphiques des fonctions de coût duales aident à visualiser les différents états des arcs décrits ci-dessus. Prenons le cas le plus simple, celui d'un arc restreint ou interdépôt. Dans ce cas, chacun des états correspond à se déplacer sur un des deux morceaux linéaires ou à se situer sur le point de brisure de la fonction de coût duale. Dans le cas d'un arc restrictif, la fonction de coût duale est linéaire en trois morceaux et par conséquent possède deux points de brisure. Ainsi, nous doublons les attributs relatifs à cet arc. Notons toutefois, qu'il n'existe que cinq états distincts. Les troisième et quatrième attributs correspondent à un état identique.

Remarquons qu'un morceau linéaire (point de brisure) de la fonction de coût primale (duale) correspond à un point de brisure (morceau linéaire) de la fonction de coût duale (primale) (voir figures 3.1 à 3.3 vs 3.4 à 3.6). Ainsi nous dirons qu'une variable restrictive est une *variable restrictive*$^+$, si elle varie sur le deuxième morceau linéaire de la fonction de coût primale ou, de manière équivalente, si $t_j^R \geq c_j^R + \frac{f_j}{a_j^R} - \varepsilon$. Dans le cas contraire, c'est-à-dire si la variable restrictive varie sur le premier morceau linéaire de la fonction de coût primale ou, de manière équivalente, si $t_j^R < c_j^R + \frac{f_j}{a_j^R} - \varepsilon$, alors nous dirons qu'il s'agit d'une *variable restrictive*$^-$. En d'autres mots, une variable restrictive$^-$ se comporte de la même manière qu'une variable restreinte. Tandis qu'une variable restrictive$^+$ correspond à violer la contrainte de capacité imposée aux variables restreintes. Cette violation est payée par le coût d'ouverture du dépôt que nous imputons alors à la variable restrictive.

Rappelons les ε–conditions de complémentarité (ε–*CC*) (Bertsekas, Hossein et Tseng, 1987) :

pour $\varepsilon > 0$ donné, $v \in R^{|A|}$ et $\pi \in R^{|N|}$ satisfont les (ε–*CC*) si et seulement si

$$v_\alpha \in [l_\alpha^\varepsilon, u_\alpha^\varepsilon] \qquad \forall\, \alpha \in A$$

où

$$l_\alpha^\varepsilon = \inf \{ \xi \mid F'_\alpha(\xi)^+ \geq t_\alpha - \varepsilon \}$$
$$u_\alpha^\varepsilon = \sup \{ \xi \mid F'_\alpha(\xi)^- \leq t_\alpha + \varepsilon \}.$$

Les bornes–ε, $[l_\alpha^\varepsilon, u_\alpha^\varepsilon]$, correspondant aux différents types d'arcs de notre problème, sont listées dans le tableau I.

α	t_α	l_α^ε	u_α^ε
arc restrictif$^-$	$t_\alpha < c_j^R - \varepsilon$	0	0
	$c_j^R - \varepsilon \le t_\alpha \le c_j^R + \varepsilon$	0	$a_j^R\ \overline{y}_j$
	$c_j^R + \varepsilon < t_\alpha < c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon$	$a_j^R\ \overline{y}_j$	$a_j^R\ \overline{y}_j$
arc restrictif$^+$	$c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \le t_\alpha \le c_j^R + \dfrac{f_j}{a_j^R} + \varepsilon$	$a_j^R\ \overline{y}_j$	a_j^R
	$t_\alpha > c_j^R + \dfrac{f_j}{a_j^R} + \varepsilon$	a_j^R	a_j^R
arc restreint client-dépôt (ijp)	$t_\alpha < c_{ijp} - \varepsilon$	0	0
	$c_{ijp} - \varepsilon \le t_\alpha \le c_{ijp} + \varepsilon$	0	$O_{ip}\ \overline{y}_j$
	$c_{ijp} + \varepsilon < t_\alpha$	$O_{ip}\ \overline{y}_j$	$O_{ip}\ \overline{y}_j$
arc restreint dépôt-client (jip)	$t_\alpha < c_{jip} - \varepsilon$	0	0
	$c_{jip} - \varepsilon \le t_\alpha \le c_{jip} + \varepsilon$	0	$D_{ip}\ \overline{y}_j$
	$c_{jip} + \varepsilon < t_\alpha$	$D_{ip}\ \overline{y}_j$	$D_{ip}\ \overline{y}_j$
arc interdépôt (kjp)	$t_\alpha < s_{kjp} - \varepsilon$	0	0
	$s_{kjp} - \varepsilon \le t_\alpha \le s_{kjp} + \varepsilon$	0	u_{kjp}
	$s_{kjp} + \varepsilon < t_\alpha$	u_{kjp}	u_{kjp}

Tableau I. Bornes–ε pour les différents types d'arcs connectés au dépôt j pour un produit p donné.

Par le corollaire de la proposition 2.2, la direction duale de descente se caractérise comme suit :

pour $\varepsilon \geq 0$, un vecteur de prix π donné et un sous-ensemble connexe de noeuds S de N, si $C_\varepsilon(S, \pi) > 0$ alors $\mu(S)$ est une direction de descente de la fonction duale Θ le long de π

où

$$C_\varepsilon(S, \pi) = \sum_{\alpha \in (S,\bar{S})} l_\alpha^\varepsilon - \sum_{\alpha \in (\bar{S},S)} \bar{u}_\alpha^\varepsilon - \sum_{i \in C_1 \cap S} O_{ip} + \sum_{i \in C_2 \cap S} D_{ip} \qquad (3.12)$$

$$\mu_i(S) = \begin{cases} -1 & \text{si } i \in S \\ 0 & \text{si } i \in \bar{S}. \end{cases}$$

Nous sommes maintenant en mesure de présenter notre algorithme, auquel nous consacrons le reste du chapitre.

III.7 Algorithme

Dans cette section, nous exposons les grandes lignes des deux versions algorithmiques mentionnées au tout début du chapitre. Nous les nommons *locale* et *globale*, selon l'instant et l'amplitude des modifications du domaine réalisable restreint.

Soit un sous-problème à résoudre en fonction d'un vecteur de coefficients d'ouverture des dépôts.

L'algorithme, inspiré principalement de *MNRELAX* est, tout comme ce dernier, paramétrisé par un scalaire positif ε. Au début de chaque itération, en plus de la paire de vecteurs flot et prix satisfaisant les (ε–*CC*), nous définissons pour chacun des dépôts une variable restrictive satisfaisant (3.7). L'algorithme s'amorce tel qu'indiqué au chapitre II, c'est-à-dire nous choisissons un noeud avec un déficit et cherchons soit une chaîne d'augmentation de flot, soit une direction duale de descente.

Dans le premier cas, nous modifions le flot sur la ε–*CAF* en permettant au flot de certains arcs (restreints) de dépasser leur capacité, sans toutefois violer la relation (3.7) qui lie ces derniers à la variable restrictive. Cela a pour but d'élargir le domaine réalisable restreint que nous pouvons alors modifier globalement ou localement.

L'approche *globale* consiste à modifier le coefficient d'ouverture d'un dépôt j dès que le flot sur un arc restreint connecté à ce dépôt viole sa contrainte de capacité. Cette façon de procéder, plutôt gloutonne, entraîne alors un ajustement de toutes les bornes–ε affectées par une telle modification et conséquemment une mise-à-jour du vecteur de flot. Ainsi, une augmentation ou une diminution du flot peut alors créer de nombreux déficits. Or, l'intérêt d'une telle approche provient du fait que nous traitions en même temps les deux problèmes : de localisation et de flot. Nous croyons, de plus, qu'elle convergera plus rapidement vers la localisation optimale des dépôts et conséquemment vers la solution optimale, puisque les coefficients d'ouverture déterminent les capacités sur le flot de certains arcs (restreints).

Dans l'approche *locale,* nous ajustons seulement les bornes–ε des arcs restreints pour lesquels il y a violation de la contrainte de capacité. Les bornes–ε de tous les autres arcs restreints demeurant inchangés, nous ne modifions donc pas les coefficients d'ouverture des dépôts à ce stade-ci. Or, nous avons, sans changer ces derniers, modifier le domaine réalisable restreint; d'où le sens d'une modification locale. A l'opposé de l'approche globale, nous traitons alternativement les deux problèmes à résoudre, en se concentrant d'abord sur celui de flot. Ainsi, l'ajustement des $\overline{y}_j$ se fera à une étape ultérieure, soit lorsqu'une solution réalisable pour le problème primal aura été trouvée.

Dans le cas où une descente duale est possible, nous effectuons une minimisation afin de déterminer le pas à faire dans la direction trouvée. S'il s'agit de la version globale, nous modifions en plus les coefficients d'ouverture des dépôts.

Dans les deux versions, après avoir exécuté un pas primal ou un pas dual, nous effectuons un changement de variables restrictives pour lesquelles il y a non-respect de (3.7). Rappelons que la variable restrictive possède une pénalité à payer lorsque la

contrainte (implicite) de capacité est violée. Or, il s'agit de choisir cette variable de telle sorte qu'une telle violation soit la moins coûteuse possible et ainsi s'assurer que la définition (3.7) soit toujours respectée.

Nous recommençons ensuite une nouvelle itération avec un autre noeud de déficit non-nul. Le processus itératif se poursuit jusqu'à ce que tous les déficits deviennent nuls, signifiant l'optimalité du sous-problème restreint considéré. Deux chemins différents se tracent alors dépendant de l'approche choisie.

Pour la version où nous modifions globalement le domaine réalisable, cela correspond également à l'optimalité de (*SRLT*), puisque nous avons résolu les deux problèmes de localisation et de flot simultanément.

Par contre, pour l'approche locale, nous devons vérifier si la distribution du flot correspond à la localisation optimale des dépôts. Cette opération est nécessitée du fait que nous avons permis la violation des contraintes de capacités, sans toutefois modifier les coefficients d'ouverture qui déterminent les capacités. Ainsi, certains dépôts initialement ouverts peuvent se fermer ou vice versa. En pratique, cela revient à ajuster les coefficients d'ouverture des dépôts, entraînant une mise-à-jour des bornes–ε, ainsi que celle du vecteur de flot de manière à satisfaire les (ε–*CC*). Si des déficits sont alors créés, cela signifie que du flot circule à travers des dépôts fermés ou que des dépôts maintenant ouverts pourraient recevoir plus de flot. Nous retournons donc résoudre un nouveau sous-problème restreint. Sinon, la solution demeurant primale-réalisable, l'optimum de (*SRLT*) est atteint et l'algorithme s'arrête.

L'organigramme (voir figure 3.7) illustre l'orchestration de *MNRELAX* avec l'algorithme de restriction. Les boîtes, ou portion de boîte, tracées en pointillé indiquent les particularités de notre méthode, tandis que les instructions encadrées de traits continus correspondent aux idées de base de l'algorithme de relaxation. Les deux versions algorithmiques se distinguent par les blocs 4 et 6.

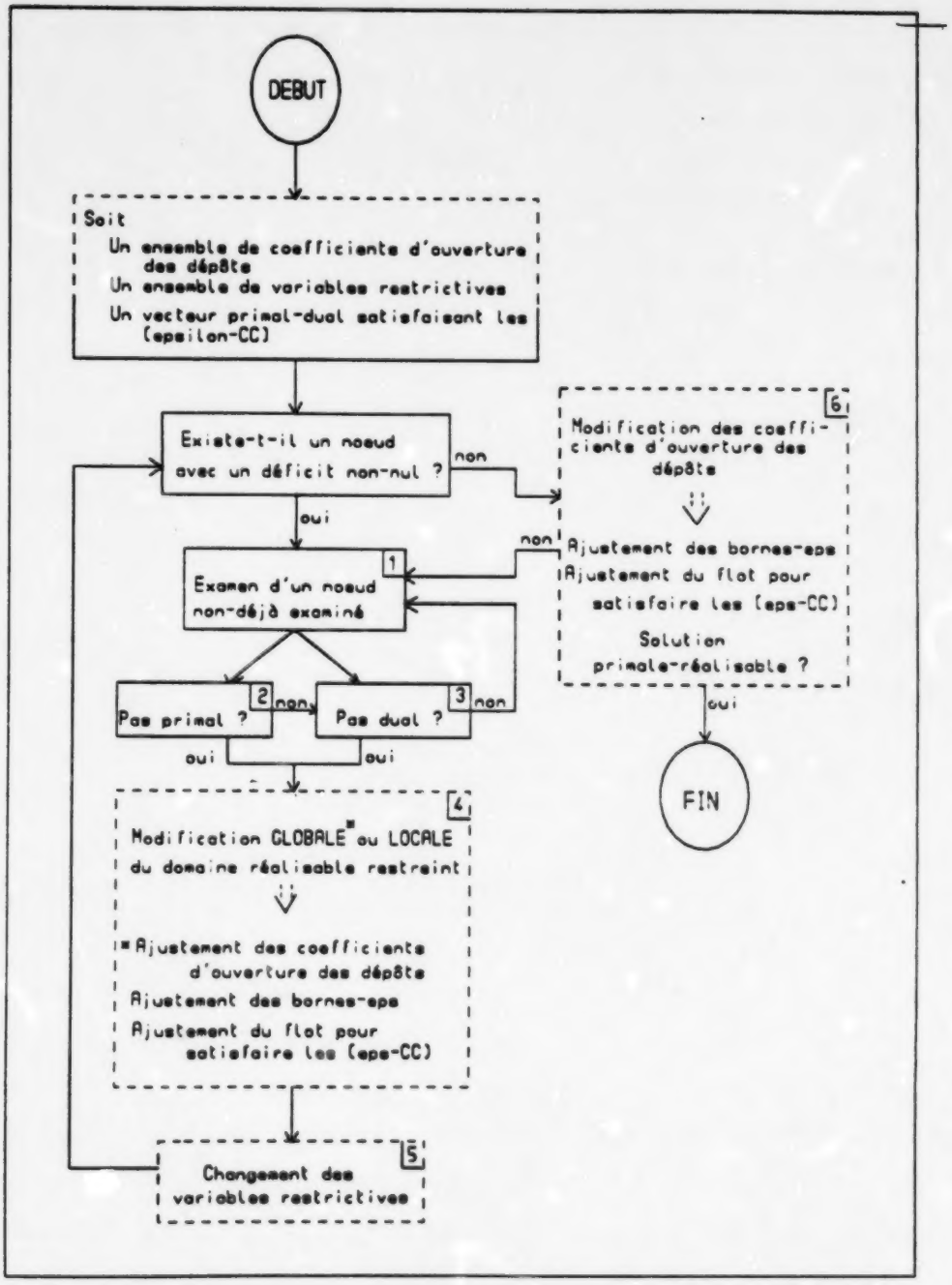


Figure 3.7. Organigramme de l'algorithme de restriction.

III.8 Particularités de l'algorithme

Dans cette section, nous détaillons les particularités de notre algorithme qui correspondent aux blocs de l'organigramme dessinés en traits pointillés. Nous explicitons, aussi, formellement les modifications apportées à chacune des étapes de *MNRELAX* qui se dissimulent dans les blocs 1 à 3 de l'organigramme.

Rappelons d'abord certaines définitions.

Soit $\bar{x}$ une valeur connue du vecteur x. Alors $\forall\, j \in D$:

$$\frac{\bar{x}_j^R}{a_j^R} \geq \begin{cases} \dfrac{\bar{x}_{ijp}}{O_{ip}} & \forall\, ijp \in R_1(j) \\ \dfrac{\bar{x}_{jip}}{D_{ip}} & \forall\, jip \in R_2(j) \end{cases} \tag{3.7}$$

$$\frac{\bar{x}_j^R}{a_j^R} \geq \frac{\bar{x}_j^m}{a_j^m} \tag{3.9}$$

$$\bar{y}_j \in \left[\frac{\bar{x}_j^m}{a_j^m}, \frac{\bar{x}_j^R}{a_j^R} \right]. \tag{3.10}$$

Dans la double perspective de clarté et de simplification, nous restreignons $\bar{y}_j$ à l'une des deux bornes définissant son intervalle de variation. En fait, la borne qui détermine le coefficient d'ouverture d'un dépôt dépend du statut de la variable restrictive relative à ce dépôt. Considérons d'abord le cas où il s'agit d'une variable restrictive$^-$. Le coefficient d'ouverture du dépôt doit être au moins égal au rapport flot/offre ou flot/demande, selon le type de l'arc, de la variable restrictive. Ceci par les définitions des bornes–ε et de la variable restrictive. Par conséquent, nous fixons $\bar{y}_j$ à sa borne supérieure. Alors que dans l'autre cas, c'est-à-dire où il s'agit d'une variable restrictive$^+$, le coefficient d'ouverture du dépôt doit être au plus égal au

rapport sus-mentionné. Afin de permettre la circulation du flot sur l'arc restrictif, nous posons $\overline{y}_j$ égal à sa borne inférieure. Donc, en résumé, chaque nouveau sous-problème est déterminé à partir des coefficients d'ouverture que nous définissons comme suit :

$$\overline{y}_j = \begin{cases} \dfrac{\overline{x}_j^m}{a_j^m} & \text{si } t_j^R \geq c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \\ \dfrac{\overline{x}_j^R}{a_j^R} & \text{si } t_j^R < c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon. \end{cases} \qquad \forall\, j \in D \qquad (3.13)$$

Cette définition favorise des bornes plus grandes sur le flot pouvant circuler sur les arcs dépôts-clients et clients-dépôts. En fait, un des buts de l'algorithme consiste justement à fournir, pour chaque sous-problème, un coefficient d'ouverture maximal pour chacun des dépôts tout en tenant compte du coût fixe à payer pour opérer ce dépôt.

De plus, définissons pour un depôt j donné :

x_{r_j} : une variable restreinte relative à ce dépôt

$$x_{r_j} \equiv \begin{cases} x_{ijp} & \text{si } r_j \equiv ijp \in R_1(j),\ j \in D \\ x_{jip} & \text{si } r_j \equiv jip \in R_2(j),\ j \in D \end{cases}$$

$$a_{r_j} \equiv \begin{cases} O_{ip} & \text{si } r_j \equiv ijp \in R_1(j),\ j \in D \\ D_{ip} & \text{si } r_j \equiv jip \in R_2(j),\ j \in D \end{cases}$$

c_{r_j} et t_{r_j} représentent respectivement le coût et la tension associés à une variable restreinte x_{r_j} du dépôt j donné.

III.8.1 Lors du marquage des noeuds

Dans cette section, nous décrivons les modifications apportées à l'étape de l'examen des noeuds de l'algorithme de relaxation.

Lors du marquage des noeuds, nous augmentons la borne–ε supérieure sur le flot de certains arcs de façon à favoriser une circulation maximale du flot. Ceci a pour avantage d'élargir le domaine réalisable et ainsi de se rapprocher du problème initial. Cette modification s'applique à certains arcs restreints qui peuvent recevoir du flot, soit ceux libres et actifs. Voyons donc jusqu'où nous pouvons repousser la borne–ε supérieure sur le flot de ces arcs et sous quelles conditions.

Distinguons les deux cas, correspondant aux deux statuts que peut prendre la variable restrictive : variable restrictive$^-$ et variable restrictive$^+$. Le raisonnement est fait pour un dépôt $j \in D$; il en est de même pour tous les dépôts.

Dans le premier cas, nous avons au début d'une itération :

$$\frac{\overline{x}_j^m}{a_j^m} \leq \frac{\overline{x}_j^R}{a_j^R} = \overline{y}_j .$$

Le coefficient d'ouverture du dépôt j définit alors les bornes–ε supérieures à la fois sur la variable restrictive et sur les variables restreintes relatives à ce dépôt.

Ainsi,

$$\frac{x_{r_j}}{a_{r_j}} \leq \overline{y}_j \qquad \forall\ r_j \in R_1(j) \cup R_2(j)$$

en particulier,

$$\frac{x_{r_j}}{a_{r_j}} \leq \overline{y}_j \qquad \forall\ r_j \in R_1(j) \cup R_2(j) \text{ t.q. } t_{r_j} \geq c_{r_j} - \varepsilon$$

$$\Rightarrow \qquad x_{r_j} \leq \overline{y}_j\ a_{r_j} \qquad \forall\ r_j \in R_1(j) \cup R_2(j) \text{ t.q. } t_{r_j} \geq c_{r_j} - \varepsilon. \qquad (3.14)$$

D'où, le terme de droite de (3.14) quantifie la valeur maximale que peut atteindre la variable restreinte libre ou active relative à j. Cette quantité correspond à la borne-ε supérieure originale.

Dans l'alternative, c'est-à-dire où il s'agit d'une variable restrictive$^+$, nous avons au début d'une itération :

$$\frac{\overline{x}_j^m}{a_j^m} = \overline{y}_j \leq \frac{\overline{x}_j^R}{a_j^R}$$

$$\Rightarrow \qquad \frac{x_{r_j}}{a_{r_j}} \leq \overline{y}_j \leq \frac{x_j^R}{a_j^R} \qquad \forall\ r_j \in R_1(j) \cup R_2(j)$$

en particulier,

$$\frac{x_{r_j}}{a_{r_j}} \leq \overline{y}_j \leq \frac{x_j^R}{a_j^R} \qquad \forall\ r_j \in R_1(j) \cup R_2(j)\ \text{ t.q. }\ t_{r_j} \geq c_{r_j} - \varepsilon. \qquad (3.15)$$

De la relation (3.15), nous constatons qu'il est possible d'éloigner la borne-ε supérieure actuelle sur le flot de l'arc restreint libre ou actif relié au dépôt j considéré.

D'où,

$$x_{r_j} \leq \frac{x_j^R}{a_j^R}\, a_{r_j} \qquad \forall\, r_j \in R_1(j) \cup R_2(j)\ \text{ t.q. }\ t_{r_j} \geq c_{r_j} - \varepsilon. \qquad (3.16)$$

L'expression définie par le membre droit de (3.16) représente donc la nouvelle borne-ε supérieure sur le flot de l'arc restreint libre ou actif, dénotée $\overline{u}_\alpha^\varepsilon$. Nous l'appelons la *borne-ε supérieure augmentée*.

De façon plus explicite,

$$\overline{u}_{\alpha}^{\varepsilon} = \begin{cases} O_{ip}\ \dfrac{x_j^R}{a_j^R} & \text{si} \begin{cases} \alpha = ijp \in R_1(j) & \text{et} \\ t_\alpha \geq c_\alpha - \varepsilon & \text{et} \\ t_j^R \geq c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \end{cases} \\ D_{ip}\ \dfrac{x_j^R}{a_j^R} & \text{si} \begin{cases} \alpha = jip \in R_2(j) & \text{et} \\ t_\alpha \geq c_\alpha - \varepsilon & \text{et} \\ t_j^R \geq c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \end{cases} \\ u_{\alpha}^{\varepsilon} & \text{sinon.} \end{cases}$$

III.8.2 Sur la chaîne d'augmentation de flot

Dans cette partie, nous explicitons les changements apportés à l'étape primale de l'algorithme de relaxation.

Enonçons, d'abord, quelques propriétés que possède la ε–*CAF* :

(*i*) l'arc restrictif, associé à un dépôt j, qui appartient à la chaîne est libre$^-$ ou libre$^+$;

(*ii*) un arc restreint qui appartient à la chaîne est libre ou actif si l'arc restrictif associé au même dépôt est libre$^+$;

(*iii*) un arc restreint qui appartient à la chaîne est libre si l'arc restrictif associé au même dépôt est libre$^-$;

(*iv*) un arc interdépôt qui appartient à la chaîne est libre.

Les trois premières propriétés découlent directement de l'extension des bornes–ε augmentées.

Déterminons maintenant la variation possible du flot sur la chaîne d'augmentation du flot.

Pour cela, nous considérons d'abord le cas le plus simple : l'arc restrictif relié au dépôt j est un arc restrictif$^-$. Dans ce cas, l'arc restrictif possède les mêmes propriétés que les autres arcs connectés au même dépôt. Par conséquent, les manipulations sur la chaîne d'augmentation de flot se font telles que décrites au chapitre II.

Cependant, dans le cas où il s'agit d'un arc restrictif$^+$, la modification classique du flot ne peut être appliquée directement. Dans ce cas, le coût fixe d'ouverture du dépôt intervient dans le calcul des bornes-ϵ sur le flot de l'arc restrictif, mais ce coût est imputé, par définition, uniquement à la variable restrictive. Ainsi, une violation de la définition de la variable restrictive ne peut être autorisée. Or, par l'extension des bornes-ϵ supérieures augmentées où le flot sur les arcs restreints est essentiellement déterminé à partir du flot sur l'arc restrictif associé au même dépôt, il peut se produire une telle dérogation.

Nous synthétisons les trois situations pouvant entraîner le non-respect de (3.7) :

1^0) $x_j^R \uparrow$ et $x_{r_j} \uparrow$ et $a_j^R \leq a_{r_j}$

2^0) $x_j^R \uparrow$ et $x_{r_j} \uparrow$ et $a_j^R > a_{r_j}$

3^0) $x_j^R \downarrow$ et $x_{r_j} \uparrow$

où x_j^R et x_{r_j} correspondent respectivement à la variable restrictive et une variable restreinte associées au dépôt j, toutes deux satisfaisant la propriété *ii*. Les symboles $\uparrow$ et $\downarrow$ signifient respectivement une augmentation et une diminution (cf. figure 3.8).

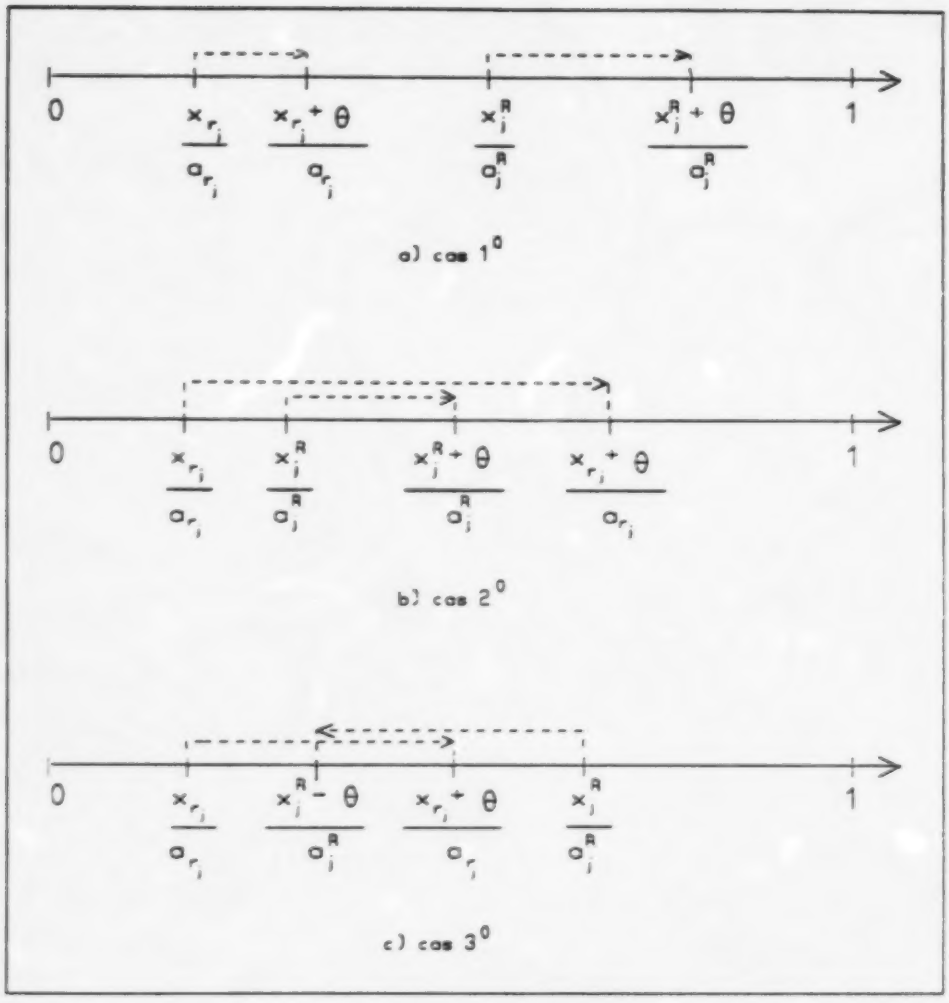


Figure 3.8. Trois cas entraînant la violation de la définition de la variable restrictive d'un dépôt j suite à une modification de flot.

Afin d'éviter une telle transgression, nous devons limiter l'augmentation ou la diminution, selon le cas, du flot pouvant circuler sur ces arcs. Il faut donc, dans les deux premiers cas, que

$$\frac{x_j^R + \theta}{a_j^R} \geq \frac{x_{r_j} + \theta}{a_{r_j}} \tag{3.17}$$

et pour le dernier cas, que

$$\frac{x_j^R - \theta}{a_j^R} \geq \frac{x_{r_j} + \theta}{a_{r_j}} \qquad \text{où } \theta \geq 0. \tag{3.18}$$

Ceci nous amène donc à calculer de nouvelles bornes-ϵ, appelées *bornes-ϵ modifiées*, qui nous empêchent d'enfreindre la définition (3.7). Le tableau II résume les diverses situations énumérées précédemment et contient les bornes-ϵ modifiées, dénotées $[\tilde{l}^{\epsilon}, \tilde{u}^{\epsilon}]$.

Par conséquent, la capacité de la chaîne (chapitre II) de longueur r, dénotée δ est déterminée par :

$$\delta = \min \{ -d_m , d_s , \delta_1 , \delta_2 , \cdots , \delta_{r-1} \}$$

où

$$\delta_\alpha = \begin{cases} \tilde{u}_\alpha^{\epsilon} - v_\alpha & \text{si } \alpha = (n_p, n_{p-1}) \\ v_\alpha - \tilde{l}_\alpha^{\epsilon} & \text{si } \alpha = (n_{p-1}, n_p) \end{cases} \qquad \forall\ p = 1, 2, \ldots, (r-1).$$

α	Cas	sens de α	$\bar{l}_\alpha^\varepsilon$	$\bar{u}_\alpha^\varepsilon$
restrictif$^+$	$x_j^R \uparrow$	$m \to s$	$l_j^{R\varepsilon}$	$\overline{u}_j^{R\varepsilon}$
restreint	$x_{r_j} \uparrow$	$m \to s$	$l_{r_j}^\varepsilon$	a_j^r
	$(a_j^R \le a_{r_j})$			
restrictif$^+$	$x_j^R \uparrow$	$m \to s$	$l_j^{R\varepsilon}$	$\min\left\{a_j^R, \dfrac{a_j^R(\overline{x}_j^R - \overline{x}_{r_j})}{(a_j^R - a_{r_j})}\right\}$
restreint	$x_{r_j} \uparrow$	$m \to s$	$l_{r_j}^\varepsilon$	$\min\left\{a_j^r, \dfrac{a_{r_j}(\overline{x}_j^R - \overline{x}_{r_j})}{(a_j^R - a_{r_j})}\right\}$
	$(a_j^R > a_{r_j})$			
restrictif$^+$	$x_j^R \downarrow$	$s \to m$	$\max\left\{a_j^R \overline{y}_j, \dfrac{a_j^R(\overline{x}_j^R + \overline{x}_{r_j})}{(a_j^R + a_{r_j})}\right\}$	$u_j^{R\varepsilon}$
restreint	$x_{r_j} \uparrow$	$m \to s$	$l_{r_j}^\varepsilon$	$\min\left\{a_{r_j}, \dfrac{a_{r_j}(\overline{x}_{r_j} + \overline{x}_j^R)}{(a_j^R + a_{r_j})}\right\}$
autres			l_α^ε	$\overline{u}_\alpha^\varepsilon$

Tableau II. Bornes–ε modifiées.

III.8.3 Pas de descente lors d'un pas dual

Nous précisons dans les prochaines lignes les éléments qui diffèrent de ceux de l'étape duale de *MNRELAX*.

Lors de la recherche du pas de descente dual, la fonction à minimiser est convexe, linéaire par morceau, et ne varie que sur la coupe. Ses points de brisure correspondent aux valeurs de λ pour lesquelles le coefficient des variables de flot de la fonction duale s'annulle. Ils sont donnés par :

$$\lambda_\alpha = \begin{cases} -\dfrac{f_j}{a_j^R} - c_j^R + t_j^R & \text{si } \alpha \text{ est un arc restrictif}^+ \\ -c_j^R + t_j^R & \text{si } \alpha \text{ est un arc restrictif}^- \\ -c_{ijp} + t_{ijp} & \text{si } \alpha = ijp \in R_1(j) \\ -c_{jip} + t_{jip} & \text{si } \alpha = jip \in R_2(j) \\ -s_{kjp} + t_{kjp} & \text{si } \alpha = kjp \end{cases} \qquad \forall\, \alpha \in [S, \bar{S}]$$

et

$$\lambda_\alpha = \begin{cases} \dfrac{f_j}{a_j^R} + c_j^R - t_j^R & \text{si } \alpha \text{ est un arc restrictif}^+ \\ c_j^R - t_j^R & \text{si } \alpha \text{ est un arc restrictif}^- \\ c_{ijp} - t_{ijp} & \text{si } \alpha = ijp \in R_1(j) \\ c_{jip} - t_{jip} & \text{si } \alpha = jip \in R_2(j) \\ s_{kjp} - t_{kjp} & \text{si } \alpha = kjp \end{cases} \qquad \forall\, \alpha \in [\bar{S}, S]$$

La modification du vecteur de prix dans la direction $\mu(S)$ se fait en ''décrémentant'' tous les prix appartenant à S d'une quantité égale à λ. Cela entraîne certains changements des bornes-ϵ qui sont illustrés dans le tableau III.

arc restrictif relatif au dépôt j				
$\alpha \in [S, \bar{S}]$	si $l_\alpha^\varepsilon = a_j^R$	et $t_j^R \le c_j^R + \dfrac{f_j}{a_j^R} + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow a_j^R \bar{y}_j$
	si $l_\alpha^\varepsilon = a_j^R \bar{y}_j$	et $t_j^R \le c_j^R + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow 0$
	si $u_\alpha^\varepsilon = a_j^R$	et $t_j^R < c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow a_j^R \bar{y}_j$
	si $u_\alpha^\varepsilon = a_j^R \bar{y}_j$	et $t_j^R < c_j^R - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow 0$
$\alpha \in [\bar{S}, S]$	si $l_\alpha^\varepsilon = 0$	et $t_j^R > c_j^R + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow a_j^R \bar{y}_j$
	si $l_\alpha^\varepsilon = a_j^R \bar{y}_j$	et $t_j^R > c_j^R + \dfrac{f_j}{a_j^R} + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow a_j^R$
	si $u_\alpha^\varepsilon = 0$	et $t_j^R \ge c_j^R - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow a_j^R \bar{y}_j$
	si $u_\alpha^\varepsilon = a_j^R \bar{y}_j$	et $t_j^R \ge c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow a_j^R$
arc restreint relatif au dépôt j				
$\alpha \in [S, \bar{S}]$	si $l_\alpha^\varepsilon = a_{\alpha_j} \bar{y}_j$	et $t_\alpha \le c_\alpha + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow 0$
	si $u_\alpha^\varepsilon = a_{\alpha_j} \bar{y}_j$	et $t_\alpha < c_\alpha - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow 0$
$\alpha \in [\bar{S}, S]$	si $l_\alpha^\varepsilon = 0$	et $t_\alpha > c_\alpha + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow a_{\alpha_j} \bar{y}_j$
	si $u_\alpha^\varepsilon = 0$	et $t_\alpha \ge c_\alpha - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow a_{\alpha_j} \bar{y}_j$
arc interdépôt relatif au produit p *(kjp)*				
$\alpha \in [S, \bar{S}]$	si $l_\alpha^\varepsilon = u_{kjp}$	et $t_\alpha \le s_\alpha + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow 0$
	si $u_\alpha^\varepsilon = u_{kjp}$	et $t_\alpha < s_\alpha - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow 0$
$\alpha \in [\bar{S}, S]$	si $l_\alpha^\varepsilon = 0$	et $t_\alpha > s_\alpha + \varepsilon$	$\Rightarrow$	$l_\alpha^\varepsilon \leftarrow u_{kjp}$
	si $u_\alpha^\varepsilon = 0$	et $t_\alpha \ge s_\alpha - \varepsilon$	$\Rightarrow$	$u_\alpha^\varepsilon \leftarrow u_{kjp}$

Tableau III. Ajustements des bornes-ε suite à une modification des prix.

III.8.4 Modifications du domaine réalisable restreint

Nous expliquons dans les pages qui suivent ce en quoi consiste le bloc 4 de l'organigramme, lequel distingue les deux versions algorithmiques.

L'extension des bornes–ε modifiées peut, suite à un pas primal, provoquer la violation des contraintes de capacité (cf. (3.5) et (3.6) dans le modèle ($SRLT(\overline{y})$)). Nous proposons alors deux approches pour modifier le domaine réalisable restreint, une locale et l'autre globale.

La première consiste à ajuster uniquement les bornes–ε des arcs restreints pour lesquels il se produit une telle violation. Par contre, l'autre modifie directement le coefficient d'ouverture du dépôt relié à un arc restreint qui viole sa contrainte de capacité, entraînant alors un changement des bornes–ε de toutes les variables restreintes relatives à ce dépôt et possiblement celle de la variable restrictive.

Dans la version globale, nous faisons en sorte que la relation (3.13) soit satisfaite au début de chacune des itérations. Tandis que dans l'autre, nous fixons $\overline{y}_j$ jusqu'à ce qu'on trouve une solution primale-réalisable en permettant toutefois à certains arcs restreints r_j, reliés au dépôt j, d'augmenter leur borne supérieure définie par $a_{r_j}\,\overline{y}_j$. D'où, le sens de modifications globale et locale du domaine réalisable restreint. L'ajustement des coefficients d'ouverture se fera, dans le deuxième cas, lorsque la réalisabilité primale sera atteinte.

III.8.5 Ajustement des coefficients d'ouverture des dépôts

L'ajustement des $\overline{y}_j$ s'effectue à différents moments dépendant de la version algorithmique considérée. Dans ce qui suit, nous décrivons en deux temps, selon qu'il s'agit de la version globale ou locale, de quelle façon et à quel moment nous faisons une telle opération.

III.8.5.1 Dans la version globale

Dans la version globale, un ajustement des coefficients d'ouverture des dépôts peut être entraîné par une modification de flot suite à un pas primal ou suite à un pas dual. De plus, seulement certains arcs qui appartiennent à la ε–*CAF* ou à la coupe, selon le cas, peuvent être affectés par un tel changement. Nous décrivons dans un pas, puis dans l'autre, les différentes situations pouvant provoquer une telle modification ainsi que les ajustements à effectuer dans chacun des cas. Les divers scénarios sont schématisés dans le tableau IV.

Suite à un pas primal :

$$\text{soit } D_1 = \left\{ j \in D \mid (i,j) \in \varepsilon\text{–}CAF,\ i \in C_1 \text{ ou } (j,i) \in \varepsilon\text{–}CAF,\ i \in C_2 \right\}.$$

Considérons d'abord le cas d'une variable restrictive⁻ associée à un dépôt $j \in D_1$. Or, par les définitions des coefficients d'ouverture des dépôts et des bornes–ε, $\overline{y}_j$ ne peut se retrouver qu'au-delà de la borne supérieure définissant son intervalle de variation (3.10). La violation de (3.13) peut alors se produire lorsque :

$$1^0) \quad \frac{x_j^R}{a_j^R} \leq \frac{x_j^m}{a_j^m} < \overline{y}_j$$

$$2^0) \quad \frac{x_j^m}{a_j^m} < \frac{x_j^R}{a_j^R} < \overline{y}_j.$$

Dans la première situation, si l'inégalité de gauche est stricte, alors outre la violation des coefficients d'ouverture des dépôts, il y a aussi violation de la définition de la variable restrictive. Afin de respecter cette dernière, nous effectuons un changement de variable restrictive qui consiste ici simplement à échanger le rôle des variables définies par x_j^R et x_j^m. Quant à l'ajustement du coefficient d'ouverture du dépôt, il se fait, dans les deux cas, en diminuant $\overline{y}_j$ à sa nouvelle borne supérieure.

Par contre, s'il s'agit d'une variable restrictive$^+$ associée au dépôt $j \in D_1$, par définition du coefficient d'ouverture du dépôt et des bornes–ε modifiées, $\overline{y}_j$ peut se retrouver en-deçà ou au-delà de sa borne inférieure, mais toujours en-deçà de sa borne supérieure. Par conséquent, nous diminuons ou augmentons $\overline{y}_j$, selon le cas, au nouveau rapport définissant sa borne inférieure. Synthétisons ces deux situations numérotées respectivement par 3^0 et 4^0 :

$$3^0) \quad \overline{y}_j < \frac{x_j^m}{a_j^m} \leq \frac{x_j^R}{a_j^R}$$

$$4^0) \quad \frac{x_j^m}{a_j^m} < \overline{y}_j \leq \frac{x_j^R}{a_j^R}.$$

Suite à un pas dual :

$$\text{soit } D_2 = \left\{ j \in D \mid (i, j) \in \text{coupe}, i \in C_1 \text{ ou } (j, i) \in \text{coupe}, i \in C_2 \right\}.$$

La modification du flot, suite à un pas dual, peut aussi entraîner la violation de (3.13) pour un dépôt $j \in D_2$. Toutefois, cela ne peut se produire que dans les premier et quatrième cas mentionnés précédemment. Le cas 2^0 est à exclure puisque dans le cas d'un arc restrictif$^-$, le flot sur cet arc peut être soit nul soit à sa borne supérieure ($a_j^R \overline{y}_j$). Par ailleurs, remarquons que le pas dual ne peut pas provoquer une ouverture des dépôts, c'est-à-dire une augmentation de $\overline{y}_j$. Cela en accord avec les définitions mêmes du coefficient d'ouverture des dépôts et des bornes–ε supérieures sur les variables restrictive et restreintes, d'où l'impossibilité du cas 3^0.

Ainsi, si un des cas 1^0 ou 4^0 survient à la suite d'un pas dual, il suffit de diminuer $\overline{y}_j$ jusqu'au nouveau rapport définissant sa borne inférieure.

Cas	pas primal	pas dual	modifications
$\frac{x_j^m}{a_j^m} > \bar{y}_j$	1. peut survenir lorsque la variable restrictive relative au dépôt j est libre* ou active*; par définition des bornes-ε supérieures modifiées	impossible puisque la borne-ε supérieure d'une variable restreinte relative au dépôt j, α_j est au plus $a_{\alpha_j}\bar{y}_j$	$\bar{y}_j \leftarrow \frac{x_j^m}{a_j^m}$ si $t_j^R \leq c_j^R + \frac{f_j}{a_j^R} + \varepsilon$ faire : $l_j^{R\varepsilon} \leftarrow a_j^R \bar{y}_j$ $\forall \alpha \in R(j) \mid t_\alpha \geq c_\alpha - \varepsilon$ faire : $u_\alpha^\varepsilon \leftarrow a_{\alpha_j}\bar{y}_j$ si $t_\alpha > c_\alpha + \varepsilon$ faire: $d_m \leftarrow d_m + (u_\alpha^\varepsilon - x_\alpha)$ $d_s \leftarrow d_s - (u_\alpha^\varepsilon - x_\alpha)$ $x_\alpha \leftarrow u_\alpha^\varepsilon$
$\frac{x_j^m}{a_j^m} < \bar{y}_j \leq \frac{x_j^R}{a_j^R}$ et $t_j^R \geq c_j^R + \frac{f_j}{a_j^R} - \varepsilon$	idem 1	idem 1	$\bar{y}_j \leftarrow \frac{x_j^m}{a_j^m}$ $l_j^{R\varepsilon} \leftarrow a_j^R \bar{y}_j$ $\forall \alpha \in R(j) \mid t_\alpha \in [c_\alpha - \varepsilon, c_\alpha + \varepsilon]$ faire: $u_\alpha^\varepsilon \leftarrow a_{\alpha_j}\bar{y}_j$
$\frac{x_j^m}{a_j^m} < \frac{x_j^R}{a_j^R} < \bar{y}_j$	2. peut survenir lorsque la variable restrictive relative au dépôt j est libre	impossible puisque la variable restrictive relative au dépôt j est soit à sa borne inférieure (= 0) soit à sa borne supérieure ($=a_j^R \bar{y}_j$)	$\bar{y}_j \leftarrow \frac{x_j^R}{a_j^R}$ $\bar{u}_j^{R\varepsilon} \leftarrow a_j^R \bar{y}_j$ $\forall \alpha \in R(j) \mid t_\alpha \in [c_\alpha - \varepsilon, c_\alpha + \varepsilon]$ faire: $\bar{u}_\alpha^\varepsilon \leftarrow a_j^\alpha \bar{y}_j$
$\frac{x_j^R}{a_j^R} \leq \frac{x_j^m}{a_j^m} < \bar{y}_j$	idem 2	idem 2	$\bar{y}_j \leftarrow \frac{x_j^m}{a_j^m}$ $\bar{u}_j^{R\varepsilon} \leftarrow a_j^R \bar{y}_j$ $\forall \alpha \in R(j) \mid t_\alpha \in [c_\alpha - \varepsilon, c_\alpha + \varepsilon]$ faire: $\bar{u}_\alpha^\varepsilon \leftarrow a_{\alpha_j}\bar{y}_j$

Tableau IV. Quatre cas nécessitant l'ajustement du coefficient d'ouverture d'un dépôt j après une modification du flot dans la version globale.

III.8.5.2 Dans la version locale

Dans la version locale, l'ajustement des coefficients d'ouverture des dépôt se fait lorsque la solution devient réalisable pour un sous-problème restreint donné. Or, il est toutefois possible que les $\overline{y}_j$ ne satisfassent plus (3.13), étant donné la permissivité accordée à certains arcs restreints de violer la contrainte de capacité. Il s'agit alors de les modifier en appliquant la définition, entraînant une mise-à-jour des bornes–ε ainsi que du vecteur flot de telle sorte que les (ε–*CC*) soient respectées. Des déficits peuvent alors être crées. Si c'est le cas, nous venons de définir un nouveau sous-problème restreint et nous recommençons le même processus itératif pour le résoudre. Sinon, nous avons trouvé la solution optimale à (*SRLT*).

III.8.6 Changement des variables restrictives

Dans cette section, nous définissons une autre étape propre à notre algorithme, soit celle du changement des variables restrictives. Dans ce qui suit, nous décrivons un tel changement pour un dépôt, il en est de même pour tous les dépôts.

Rappelons que le flot sur les arcs restreints connectés à un dépôt j est essentiellement déterminé par le flot sur l'arc restrictif relatif à ce dépôt. Cela découle des définitions des bornes–ε modifiées et du coefficient d'ouverture du dépôt, selon qu'il s'agit d'un arc restrictif$^+$ ou restrictif$^-$. En plus, la variable restrictive, par définition, ne possède pas de contrainte explicite de capacité. Nous lui affectons toutefois une pénalité, correspondant au coût d'ouverture du dépôt.

Le choix de cette variable se fait donc de façon à assurer des capacités optimales sur le flot des arcs clients-dépôts et dépôts-clients, tout en tenant compte du coût fixe du dépôt. Il s'agit de sélectionner, parmi tous ces arcs, celui pour lequel il est avantageux de pousser plus de flot, c'est-à-dire sans qu'il en coûte trop cher de violer la contrainte de capacité.

Ainsi, un changement de la variable restrictive, pour un dépôt j, s'effectue lorsque $\frac{x_j^R}{a_j^R} \leq \frac{x_j^m}{a_j^m}$. Autrement, la variable définissant x_j^R satisfait déjà la définition de

la variable restrictive et remplit donc tous les critères mentionnés ci-dessus.

En pratique, comment s'effectue ce changement ?

Se référant au tableau I, nous pouvons remarquer que : plus la tension est grande par rapport au coût c_α, plus le flot sur cet arc augmente. Or, par définition, la variable restrictive correspond à la variable parmi toutes les variables clients-dépôts et dépôts-clients qui fournit le plus grand ratio flot/offre ou flot/demande pour un dépôt j et pour laquelle la violation de la contrainte de capacité est la moins coûteuse. Les critères du choix de la variable restrictive d'un dépôt j reposent donc sur le comportement des variables duales aux noeuds de l'arc par rapport au coût de l'arc.

Nous proposons trois critères pour sélectionner la variable restrictive associée à un dépôt j. Chacun de ces critères est défini sur une partition disjointe de l'ensemble des arcs restreints relatifs au dépôt j. Chacune de ces partitions regroupe les arcs restreints les plus susceptibles de devenir l'arc restrictif. Par exemple, une première partition correpond à l'ensemble constitué de tous les arcs restreints actifs. Un autre ensemble regroupe les arcs restreints libres saturés. Puis finalement, le dernier ensemble réunit tous les autres arcs restreints.

Soit

$$P^+(j) = \{\, \alpha \in R_1(j) \cup R_2(j) \mid t_\alpha > c_\alpha + \varepsilon \,\}$$
$$P(j) = \{\, \alpha \in R_1(j) \cup R_2(j) \mid c_\alpha - \varepsilon \le t_\alpha \le c_\alpha + \varepsilon \text{ et } x_\alpha = u_\alpha^\varepsilon \,\}$$
$$P^-(j) = \{\, \alpha \in R_1(j) \cup R_2(j) \mid \alpha \in P^+(j) \cup P(j) \,\}.$$

Afin de déterminer la variable restrictive relative à un dépôt j, nous considérons d'abord comme arcs restrictifs potentiels, les arcs appartenant à $R^+(j)$. Si cet ensemble est vide, nous examinons les arcs appartenant à $R(j)$. Si cet ensemble est également vide, alors nous travaillons sur le dernier ensemble $R^-(j)$. Dépendant de l'ensemble non-vide, nous définissons différents critères, résumés dans le tableau V, pour choisir la variable restrictive .

Condition	Critère du choix de la variable restrictive	Critère du changement
$P^+(j) \neq \emptyset$	Soit x_j^n la variable définie par $\Delta_j^+ = \max_{\alpha_j \in P^+(j)} \left\{ \frac{t_{\alpha_j}}{c_{\alpha_j} + \frac{f_j}{a_{\alpha_j}} - \varepsilon} \right\}$	$\Delta_j^+ > \left\{ \frac{t_j^R}{c_j^R + \frac{f_j}{a_j^R} - \varepsilon} \right\}$ ou $t_j^R \leq c_j^R + \varepsilon$
$P^+(j) = \emptyset$ $P(j) \neq \emptyset$	Soit x_j^n la variable définie par $\Delta_j = \min_{\alpha_j \in P(j)} \left\{ c_{\alpha_j} + \frac{f_j}{a_{\alpha_j}} \right\}$	$\Delta_j < c_j^R + \frac{f_j}{a_j^R}$ et $t_j^R \in [c_j^R - \varepsilon, c_j^R + \varepsilon]$ ou $\frac{x_j^R}{a_j^R} < \frac{x_j^m}{a_j^m}$ ou $t_j^R < c_j^R - \varepsilon$
$P^+(j) = \emptyset$ $P(j) = \emptyset$	Soit x_j^n la variable définie par $\Delta_j^- = \min_{\alpha_j \in P^-(j)} \left\{ c_{\alpha_j} + \frac{f_j}{a_{\alpha_j}} \right\}$	$\Delta_j^- < c_j^R + \frac{f_j}{a_j^R}$ et $t_j^R < c_j^R - \varepsilon$

Tableau V. Trois critères de sélection de la variable restrictive relative à un dépôt j donné.

Nous avons mentionné, au début de cette section, que nous changeons la variable restrictive d'un dépôt j seulement lorsque $x_j^m/a_j^m \geq x_j^R/a_j^R$. Toutefois, puisque ce sont les variables duales qui déterminent les variables restrictives, les variables duales demeurant inchangées suite à un pas primal, nous effectuons un tel changement seulement si $x_j^R/a_j^R < x_j^m/a_j^m$. Il suffit alors d'échanger le statut des variables x_j^R et x_j^m.

III.9 Conclusion

Pour terminer, esquissons les différences entre les deux approches : globale et locale, tout en soulignant les mérites de chacune. Les deux versions se caractérisent par la façon dont s'effectue le changement du domaine réalisable restreint. Rappelons que celui-ci est principalement déterminé par les capacités, lesquelles sont définies à partir des coefficients d'ouverture des dépôts. L'attrait majeur de l'algorithme proposé découle directement des nouvelles formulations mathématiques dérivées de la relaxation forte du problème qui nous permettent de traiter comme un seul problème le problème de localisation ainsi que celui de flot.

Ainsi, dans la version globale, aussi qualifiée de gloutonne, nous apportons des modifications majeures au domaine réalisable restreint. Cela se fait en changeant la valeur du coefficient d'ouverture des dépôts pour lesquels nous avons augmenté ou diminué le flot sur des arcs restreints ou de l'arc restrictif reliés à ce dépôt. Cette façon de procéder entraîne alors un ajustement des bornes-ϵ, ainsi que des flots sur les dits-arcs rattachés au dépôt. Cela peut évidemment créer de nombreux déficits, mais nous croyons converger plus rapidement vers la solution optimale en permettant la modification au fur et à mesure des coefficients d'ouverture. L'intérêt d'une telle approche vient du fait que nous résolvons simultanément les problèmes de localisation et de flot.

A l'opposé, l'approche locale, optimise d'abord le flot pour un vecteur donné d'ouverture des dépôts, en permettant toutefois à certains arcs de dépasser leur capacité. En effet, après une modification du flot, nous n'ajustons que les capacités sur les

arcs pour lesquels cette contrainte est violée, assurant ainsi une diminution du déficit total, ce qui n'est pas le cas pour la version globale. Or, nous attendons d'atteindre la réalisabilité primale avant d'ajuster les coefficients d'ouverture des dépôts et ainsi vérifier si la localisation des dépôts est optimale. Si ce n'est pas le cas, nous définissons un nouveau domaine réalisable restreint et retournons résoudre un nouveau sous-problème. Cette approche a l'avantage de fournir un meilleur contrôle sur le comportement de l'algorithme, puisqu'elle possède beaucoup moins de variabilité que la précédente.

Nous avons programmé une seule des deux versions algorithmiques, soit la version globale. C'est donc à partir de celle-ci que nous effectuons les expérimentations numériques qui servent à valider notre méthode. Nous présentons les résultats numériques obtenus dans le prochain chapitre.

CHAPITRE IV

EXPERIMENTATIONS NUMERIQUES

Nous avons programmé en FORTRAN 77 une version de l'algorithme de restriction présentée au chapitre précédent pour résoudre le problème transformé de la relaxation forte du problème de localisation-distribution avec transferts interdépôts. Il s'agit de la version gloutonne où nous modifions de façon globale les coefficients d'ouverture des dépôts. Une disquette du programme est disponible sur demande. L'autre version discutée est en voie d'implantation. Les expérimentations numériques ont été effectuées sur des stations SUN/3, équipées d'un microprocesseur MC68020 et d'un coprocesseur mathématique MC68881 opérant à 15*Mhz* sous le système d'exploitation UNIX.

Ce chapitre se développe de la façon suivante. Tout d'abord, nous décrivons brièvement les problèmes qui ont servi à tester notre algorithme, puis nous donnons les étapes d'initialisation ainsi que celles de la version algorithmique programmée, nous mentionnons certains détails d'implantation, et nous présentons finalement les résultats obtenus accompagnés de commentaires.

IV.1 Problèmes-test

Les problèmes utilisés pour tester notre algorithme proviennent d'une série de problèmes générés aléatoirement (voir appendice C pour la description de la génération de ces problèmes). L'échantillon-test se compose de 27 problèmes. Chacun d'eux comporte 50 noeuds et est dénoté par le code PZ#P$. Le premier P représente la taille du problème, P pour petit. La notation Z# correspond au facteur zone qui sert à simuler le déséquilibre dans le réseau; nous en définissons 3 (#=1,2,3). Le sigle P$ indique la présence des divers produits; nous en comptons au plus 9 ($=1-6,8,9). Le tableau VI contient les caractéristiques des problèmes-test utilisés.

Problème	# clients *	# dépôts	# produits	# total d'arcs
PZ*P1	42,42,41	8,8,9	1	728,728,810
PZ*P2	42,41,41	8,9,9	2	1456,1620,1620
PZ*P3	42,41,41	8,9,9	3	2184,2430,2430
PZ*P4	42,41,41	8,9,9	4	2912,3240,3240
PZ*P5	42,41,41	8,9,9	5	3640,4050,4050
PZ*P6	42,42,41	8,8,9	6	4368,4368,4860
PZ*P7	42,41,42	8,9,8	7	5096,5670,5096
PZ*P8	48,41,41	8,9,9	8	5824,6480,5096
PZ*P9	42,41,41	8,9,9	9	6552,7290,7290

Tableau VI. Caractéristiques de l'échantillon-test où * = 1,2,3.

Pour la présente étude, les tests numériques se sont limités à des problèmes de petite taille pour lesquels l'algorithme du "branch-and-bound" (Hodgson, 1988) nous fournit la solution optimale, ceci afin de déterminer la validité de notre algorithme ainsi que la qualité de la solution qu'il génère.

IV.2 Programme

Lors d'une première phase du développement du programme, nous n'avons pas insisté sur l'optimisation de la codification de l'algorithme puisque d'une part, nous possédions le programme-source *MNRELAX* et d'autre part, le but que nous visions consistait, d'abord, à valider la méthode proposée dans cette étude. Ainsi, nous avons extrait certaines parties du programme de l'algorithme de relaxation que nous avons greffées au nôtre. Par conséquent, nous ne discutons ni des structures de données utilisées, ni des alternatives d'implantation qui relèvent directement de l'algorithme de relaxation. Une discussion de ces sujets, relatifs à *MNRELAX*, se retrouve dans la thèse de Tseng (1986). En ce qui nous concerne, cela constitue une étape ultérieure à nos travaux. Dans cette section, nous donnons une procédure d'initialisation, ainsi que les étapes d'une itération de la version algorithmique programmée en plus de souligner certains détails d'implantation de l'algorithme de restriction.

IV.2.1 Version globale

Nous présentons une procédure d'initialisation ainsi que les étapes d'une itération de l'algorithme de restriction, construite à partir d'un noeud ayant un déficit positif. Nous référons le lecteur à l'appendice D pour une itération faite sur un noeud avec un déficit négatif. Des variantes d'initialisation seront discutées à la section suivante.

Etape d'initialisation – *INIT* 0

Poser $x = 0$, $w = 0$. Ceci implique, par (3.11), que :

$$d_i = \begin{cases} -O_{ip} & \forall\, i \in C_1 \\ D_{ip} & \forall\, i \in C_2 \end{cases} \qquad \forall\, p \in P$$

$$d_j = 0 \qquad \forall\, j \in D.$$

Poser aussi $\bar{y} = 0, \quad \pi = 0, \quad l^{\varepsilon} = 0, \quad u^{\varepsilon} = 0.$

Considérer comme variable restrictive associée au dépôt j, la variable x_j^R t.q. :

$$\min \left\{ \min_{\substack{i \in C_1 \\ p \in P}} \left\{ c_{ijp} + \frac{f_j}{O_{ip}} \right\}, \min_{\substack{i \in C_2 \\ p \in P}} \left\{ c_{jip} + \frac{f_j}{D_{ip}} \right\} \right\} \quad \forall\, j \in D.$$

Itération de l'algorithme

Etape 1. Choisir un noeud s *t.q.* $d_s > 0$.

S'il n'existe pas de tel noeud, alors *STOP*.

Sinon faire : $M = \{s\}$

$S = \varnothing$

où

M = ensemble des noeuds étiquetés, non-examinés

S = ensemble des noeuds étiquetés et examinés.

Aller à l'étape 2.

Etape 2. Choisir un noeud $k \in M$.

Faire : $M \leftarrow M - \{k\}$

$S \leftarrow S \cup \{k\}$.

Aller à l'étape 3.

Etape 3. Examiner le noeud k en donnant l'étiquette k et en marquant tous les noeuds non étiquetés $m \in \Gamma(k)$ de la manière suivante :

$$M \leftarrow M \cup \{m\} \quad \text{si} \quad \begin{cases} \alpha = (k, m) \in A \mid v_\alpha > l_\alpha^{\varepsilon} \\ \text{ou} \\ \alpha = (m, k) \in A \mid v_\alpha < \bar{u}_\alpha^{\varepsilon} \end{cases}$$

où

$$\bar{u}_\alpha^\varepsilon = \begin{cases} O_{ip}\,\dfrac{x_j^R}{a_j^R} & \text{si} \begin{cases} \alpha = ijp \in R_1(j) & \text{et} \\ t_\alpha \geq c_\alpha - \varepsilon & \text{et} \\ t_j^R \geq c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \end{cases} \\ D_{ip}\,\dfrac{x_j^R}{a_j^R} & \text{si} \begin{cases} \alpha = jip \in R_2(j) & \text{et} \\ t_\alpha \geq c_\alpha - \varepsilon & \text{et} \\ t_j^R \geq c_j^R + \dfrac{f_j}{a_j^R} - \varepsilon \end{cases} \\ u_\alpha^\varepsilon & \text{sinon.} \end{cases}$$

Si $C_\varepsilon(S, \pi) > 0$ ou si $|S|$ = nombre de noeuds examinés, aller à l'étape 6

où

$$C_\varepsilon(S, \pi) = \sum_{\alpha \in (S, \bar{S})} l_\alpha^\varepsilon - \sum_{\alpha \in (\bar{S}, S)} \bar{u}_\alpha^\varepsilon - \sum_{i \in C_1 \cap S} O_{ip} + \sum_{i \in C_2 \cap S} D_{ip}.$$

Sinon,

s'il existe un noeud $m \in M$ étiqueté à partir de k *t.q.* $d_m < 0$, aller à l'étape 4. Sinon retourner à l'étape 2.

Etape 4. Retracer la ε–*CAF* qui commence au noeud m (avec $d_m < 0$) et se termine au noeud s. Soit δ la capacité de la chaîne. Incrémenter de δ le flot sur les arcs orientés de m vers s et décrémenter de δ le flot sur les arcs orientés dans le sens inverse. Ajuster les déficits aux extrémités de la ε–*CAF*. Aller à l'étape 5.

Etape 5. Mettre à jour les coefficients d'ouverture des dépôts $\forall\, j \in D_1$

où

$$D_1 = \left\{ j \in D \mid (i, j) \in CAF, i \in C_1 \text{ ou } (j, i) \in CAF, i \in C_2 \right\}.$$

Ajuster les bornes–ε. Effectuer un changement des variables restrictives s'il y a lieu. Passer à l'itération suivante.

Etape 6. Rechercher le pas dual λ^* t.q. :

$$\Theta(\pi + \lambda^* \mu(S)) = \min_{\lambda > 0} \left\{ q(\pi + \lambda \mu(S)) \right\}$$

où

$$\mu_i(S) = \begin{cases} -1 & \text{si } i \in S \\ 0 & \text{si } i \in \overline{S}. \end{cases}$$

Faire $\pi \leftarrow \pi + \lambda^* \mu(S)$. Ajuster les bornes–ε et le vecteur flot afin de satisfaire les (ε–*CC*). Aller à l'étape 7.

Etape 7. Ajuster les coefficients d'ouverture des dépôts $\forall\, j \in D_2$

où

$$D_2 = \left\{ j \in D \mid (i, j) \in \text{coupe}, i \in C_1 \text{ ou } (j, i) \in \text{coupe}, i \in C_2 \right\}.$$

Faire un changement de variables restrictives s'il y a lieu.

Passer à l'itération suivante.

IV.2.2 Implantation de l'algorithme

Dans cette sous-section, nous décrivons certains détails d'implantation, soient ceux relatifs au calcul de la quantité $C_\varepsilon(S, \pi)$, ainsi qu'à la recherche du pas optimal de descente lors d'un pas dual. Nous signalons également la nécessité d'imposer des capacités et énonçons le critère d'arrêt de l'algorithme.

De façon analogue à ce qui a été fait dans l'algorithme de relaxation primal-dual (Tseng, 1986), nous utilisons une expression alternative à celle décrite dans le chapitre III pour évaluer $C_\varepsilon(S, \pi)$ (cf. section III.6).Ceci dans le but de simplifier le calcul de cette quantité. Dans ce qui suit, nous dérivons la formule employée. Nous conservons les mêmes notations que celles introduites à la section III.6.

Soit une paire primale-duale quelconque (v, π) satisfaisant les (ε–*CC*). Récrivons la définition (3.12) :

$$C_\varepsilon(S, \pi) = \sum_{\alpha \in (S, \bar{S})} l_\alpha^\varepsilon - \sum_{\alpha \in (\bar{S}, S)} \bar{u}_\alpha^\varepsilon - \sum_{i \in C_1 \cap S} O_{ip} + \sum_{i \in C_2 \cap S} D_{ip}$$

$$= \sum_{\alpha \in (S, \bar{S})} (l_\alpha^\varepsilon - v_\alpha) + \sum_{\alpha \in (\bar{S}, S)} (-\bar{u}_\alpha^\varepsilon + v_\alpha) + \sum_{\alpha \in (S, \bar{S})} v_\alpha - \sum_{\alpha \in (\bar{S}, S)} v_\alpha$$
$$- \sum_{i \in C_1 \cap S} O_{ip} + \sum_{i \in C_2 \cap S} D_{ip}$$

$$= \sum_{\alpha \in (S, \bar{S})} (l_\alpha^\varepsilon - v_\alpha) - \sum_{\alpha \in (\bar{S}, S)} (\bar{u}_\alpha^\varepsilon + v_\alpha)$$
$$+ \Big(\sum_{\substack{\alpha \in (S, \bar{S}) \\ \alpha = ijp}} v_\alpha - \sum_{i \in C_1 \cap S} O_{ip} \Big) + \sum_{\substack{\alpha \in (S, \bar{S}) \\ \alpha \neq ijp}} v_\alpha$$
$$+ \Big(- \sum_{\substack{\alpha \in (\bar{S}, S) \\ \alpha = jip}} v_\alpha + \sum_{i \in C_2 \cap S} D_{ip} \Big) - \sum_{\substack{\alpha \in (\bar{S}, S) \\ \alpha \neq jip}} v_\alpha$$

où $\alpha = ijp$ signifie que l'arc α est de la forme client-dépôt, tandis que $\alpha \neq ijp$ correspond à arc de la forme dépôt-dépôt ou dépôt-client. Même chose lorsque nous sommons sur $(S, \bar{S})$: $\alpha = jip$ signifie que l'arc α est de la forme dépôt-client, et

$\alpha \neq jip$ correspond à une arc dépôt-dépôt ou client-dépôt.

Soit d le vecteur déficit correspondant au vecteur v. Par définition du déficit (cf. équation (3.11)), nous avons que : la somme des déficits aux noeuds dans S égal au flot net du sous-ensemble S, c'est-à-dire :

$$C_\varepsilon(S, \pi) = -\left\{ \sum_{\alpha \in (S,\bar{S})} (v_\alpha - l_\alpha^\varepsilon) + \sum_{\alpha \in (\bar{S},S)} (\bar{u}_\alpha^\varepsilon - v_\alpha) \right\} + \sum_{i \in S} d_i \qquad (4.1)$$

Ainsi, $C_\varepsilon(S, \pi)$ se calcule plus facilement lorsque la formule (4.1) est utilisée plutôt que celle définie par (3.12). Cette expression permet aussi une mise-à-jour récursive de $C_\varepsilon(S, \pi)$ (cf. Tseng, 1986 pour une discussion plus détaillée).

Le pas optimal de descente dans la direction $\mu(S)$ le long de π est rencontré en un des points de brisure de la fonction duale (convexe, linéaire par morceau). Ainsi, nous trions, à l'aide d'un "heapsort", tous les points de brisure et cherchons parmi eux, en effectuant une recherche dichotomique, celui pour lequel la fonction duale est minimisée. En pratique, nous trions seulement les points de brisure associés aux arcs qui appartiennent à la coupe correspondant à la direction de descente duale.

Un autre point qui nécessite d'être mentionné concerne les capacités sur les arcs. Rappelons que dans le modèle restreint les variables associées aux arcs interdépôts ne possèdent pas de capacités. Or, nous savons que l'algorithme de relaxation se comporte beaucoup mieux lorsque les capacités sur les arcs sont serrées. Dans cette perspective, nous définissons, pour les arcs interdépôts, une borne supérieure égale à la somme des offres pour un produit p donné, soit $u_{kjp} = \sum_{i \in C_1} O_{ip}$.

D'autre part, le critère d'arrêt de l'algorithme consiste à évaluer l'écart relatif entre les coûts primal et dual une fois tous les déficits annullés, c'est-à-dire lorsque la solution devient primale-réalisable. Ainsi, l'algorithme se termine selon un degré de précision défini à priori, c'est-à-dire lorsque la relation suivante est satisfaite :

$$\frac{solution\,(Primal) - solution\,(Dual)}{solution\,(Primal)} < pr\acute{e}cision.$$

Si cette relation est vérifiée, l'optimalité est atteinte, sinon nous diminuons la valeur de ε, recalculons les bornes–ε et retournons au début de l'algorithme recommencer un deuxième cycle. Au cours de celui-ci nous cherchons uniquement à redistribuer le flot optimalement; les coefficients d'ouverture des dépôts étant optimaux. Nous fixons donc les $\overline{y}_j$.

IV.3 Résultats et commentaires

Outre la procédure d'initialisation décrite au chapitre précédent, INIT0, nous en avons programmé et testé deux autres. La première, INIT1, à l'opposé de celle déjà présentée, consiste à ouvrir tous les dépôts et trouver le flot à coût minimum par l'algorithme de relaxation pour coûts linéaires. Tandis que dans la seconde, INIT2, nous sélectionnons les dépôts à ouvrir en affectant l'offre et la demande de chaque client pour chacun des produits au dépôt le moins coûteux. Ce coût fait intervenir le coût de transport ainsi que celui d'opération et d'ouverture du dépôt tout en tenant compte de la proportion entre ce dernier et l'offre ou de la demande du client. Il s'agit donc, pour un client i et un produit p donnés, d'affecter l'offre ou la demande correspondante au dépôt j' satisfaisant :

$$\min_{j \in D} \left\{ c_{ijp} + \frac{f_j}{O_{ip}} \right\} \quad \text{si } i \in C_1$$

$$\min_{j \in D} \left\{ c_{jip} + \frac{f_j}{D_{ip}} \right\} \quad \text{si } i \in C_2.$$

En pratique, cela revient à ouvrir une proportion quand même considérable de dépôts, soit 90%. Ceci, étant donné les coûts d'ouverture relativement petits par rapport aux coûts de transport dans les problèmes générés aléatoirement. Nous avons pu constaté qu'une ouverture des dépôts lors de l'initialisation engendre une borne supérieure moins bonne que celle générée par une fermeture des dépôts. Cela

s'explique par le coût faible d'opération du dépôt qui ne favorise pas une fermeture du dépôt une fois ouvert. Ceci a pour effet de limiter l'exploration du domaine réalisable. Nous avons donc ignoré la procédure INIT1 qui détermine une solution de départ en fixant tous les dépôts ouverts, puisqu'une telle initialisation génère une borne supérieure significativement moins bonne que les deux autres. Ainsi, seules INIT0 et INIT2 serviront à générer une solution initiale pour tester l'algorithme de restriction.

Les expérimentations numériques faites avec INIT0 fournissent des solutions seulement pour les problèmes comportant au plus 3 produits. Tandis qu'en exécutant notre programme avec INIT2, nous résolvons des problèmes contenant jusqu'à neuf produits. Ainsi, nous divisons les résultats numériques obtenus en deux tableaux. Le premier, numeroté VII, résume ceux générés avec les deux procédures d'initialisation. Tandis que l'autre, le tableau VIII, contient les solutions des problèmes de plus de quatre produits.

Tous les résultats sont comparés à la solution optimale que nous procure l'algorithme du "branch-and-bound" (Hodgson, 1988), en calculant l'écart relatif de la solution obtenue par rapport à la solution optimale, soit :

$$écart = \frac{(\ solution\,(INITx) - solution\ \ optimale\ )}{solution\,(optimale)}\ \% \qquad \text{où } x = 0, 2.$$

L'astérisque, (*), dans le tableau VII indique une solution non-entière. Cela survient à deux reprises seulement. Tandis que la croix, (†), que nous retrouvons dans les deux tableaux, est associée à une solution optimale.

Rappelons qu'étant donné l'utilisation d'une partie de code déjà existant, nous ne nous sommes pas concentrés sur une programmation optimale de l'algorithme. C'est la raison pour laquelle nous ne rapportons aucun temps d'exécution dans les résultats numériques. Cependant, nous tiendrons compte de ce facteur dans une phase ultérieure au développement de l'algorithme.

Problème-test	INIT0 (écart (%))	INIT2 (écart (%))	solution optimale "branch-and-bound"
PZ1P1	1 458 374, (0,4)	1 453 602, (0,08)	1 452 422,
PZ2P1	6 797 041,* (0,01)	6 801 221, (0,07)	6 796 241,
PZ3P1	7 284 593,* (0,05)	7 287 140, (0,09)	7 280 777,
PZ1P2	8 359 718,† (0,01)	8 365 828, (0,08)	8 358 953,
PZ2P2	14 153 280,† (0,01)	14 153 280,† (0,01)	14 151 752,
PZ3P2	16 889 466,† (0,01)	16 889 466,† (0,01)	16 887 898,
PZ1P3	17 507 374,† (0,01)	17 507 374,† (0,01)	17 505 704,
PZ2P3	21 654 428,† (0,01)	21 654 428,† (0,01)	21 650 954,
PZ3P3	20 516 964,† (0,01)	20 516 964,† (0,01)	20 514 330,

Tableau VII. Tableau des résultats pour 1,2 et 3 produits.

Problème-test	INIT2 (écart (%))	solution optimale "branch-and-bound"
PZ1P4	26 306 522,' (0,01)	26 302 948,
PZ2P4	31 127 796,' (0,01)	31 168 944,
PZ3P4	22 841 636,' (0,02)	22 836 992,
PZ1P5	26 124 572,' (0,01)	26 121 346,
PZ2P5	32 799 536,' (0,01)	32 795 508,
PZ3P5	39 410 968,' (0,01)	39 405 696,
PZ1P6	40 408 636, (2,4)	39 476 128,
PZ3P6	35 161 052,' (0,02)	35 155 692,
PZ1P8	46 894 752,' (0,02)	46 885 972,
PZ2P8	54 583 632,' (0,2)	54 476 304,
PZ3P8	42 736 232,' (0,02)	42 728 600,
PZ2P9	53 527 068,' (0,02)	53 517 968,

Tableau IX. Tableau des résultats pour 4 produits et plus.

Nous pouvons remarquer qu'une solution de départ générée avec INIT0 produit une borne supérieure de qualité égale ou supérieure à celle obtenue en utilisant INIT2. Cela s'explique par la remarque faite précédemment. Dans le pire cas, la borne supérieure générée par notre algorithme pour la série de problèmes de 3 produits et moins se situe à 0,4% de la solution optimale. Tandis que dans le meilleur cas, qui correspond à l'optimum, l'écart passe à 0,01% et survient pour les deux tiers de ces problèmes. Cet écart non-nul origine probablement d'une accumulation d'erreurs numériques.

Nous observons une légère déterioration du comportement de l'algorithme pour les problèmes de plus de 3 produits (cf. tableau VIII). En effet, seulement le deux tiers des problèmes composant cette deuxième série de problèmes-test génère une borne supérieure se situant en-deça de 2,4% de la solution optimale. Parmi ces derniers, une seule cependant n'est pas optimale. Dans tous les autres cas, non-représentés dans le tableau, l'algorithme ne semble pas converger. Cela se produit pour les problèmes suivants : PZ2P6, PZ*P7, *=1,2,3 et PZ*P9, *=1,3.

Quelles conclusions pouvons-nous tirer de ces premières expérimentations ?

Tout d'abord, comme nous l'avons déjà mentionné, la solution initiale affecte la convergence de l'algorithme. En ce qui concerne les variantes d'initialisation testées jusqu'ici, combinées aux problèmes actuels, cela est principalement causé par les coûts fixes trop petits relativement aux coûts de transport. Or, lors d'éventuelles expérimentations, nous pourrions pondérer les coûts par un facteur par exemple, égal à 100, de façon à favoriser une discrimination des deux sortes de coûts présents dans le modèle. D'autre part, il pourrait être intéressant de tester d'autres variantes. Par exemple, n'ouvrir qu'une proportion des dépôts présentement ouverts par INIT2.

D'autre part, la version qui a servi aux tests possède beaucoup de variabilité dû principalement aux nombreux changements de domaines réalisables qui croissent avec la taille du problème. D'où la non-convergence pour certains problèmes avec INIT2 ainsi que pour les problèmes de plus de 3 produits avec INIT0. Ceci peut s'expliquer du fait que plus la taille d'un problème est grande, plus l'algorithme se promène dans

le domaine réalisable risquant ainsi de revenir à plus d'une reprise dans un même domaine réalisable restreint et prendre beaucoup plus de temps avant d'atteindre l'optimalité. Intuitivement, ce phénomène pourrait être comparé à un effet de "zigzagage" ou même dans certains cas, cela pourrait créer un effet cyclique. Ainsi, plus le nombre de modifications globales du domaine réalisable, directement lié à la taille du problème, augmente, plus la convergence de l'algorithme devient difficile à prévoir. Une façon de contôler le comportement de l'algorithme consiste à diminuer la fréquence de tels changements. Il s'agit de la deuxième version discutée dans ce mémoire.

Un dernier point à souligner relève de l'algorithme de relaxation primal-dual. Il s'agit des capacités sur le flot des arcs interdépôts. Nous croyons que les bornes actuelles, égales à $[0, \sum_{i \in C_1} O_{ip}]$, peuvent engendrer d'importants déficits qui peuvent devenir impossibles à éliminer, rendant le problème irréalisable. Ainsi, il pourrait être bénéfique de resserrer davantage ces bornes.

En conclusion, la version de l'algorithme de restriction que nous avons programmée génère de bons résultats, voire optimaux, pour la grande majorité des problèmes testés. Malgré la non-convergence pour certains problèmes, ces premières expérimentations numériques se révèlent tout de même encourageantes et motivent la poursuite de nos travaux. En effet, nous pouvons comprendre que la version implantée complique la convergence de l'algorithme étant donné la grande part de variabilité qu'elle possède. Or, nous croyons que la seconde version constituera une panacée à ce problème.

CONCLUSION

Le but de ce mémoire était de développer une méthode de résolution exacte ou heuristique pour résoudre le problème de localisation-distribution avec échanges interdépôts. Rappelons le contexte de la problématique de résolution.

La formulation initiale du problème se révèle complexe à résoudre. Or, les contraintes d'équilibrage confèrent au modèle une structure de réseau bénéfique aux développements algorithmiques. C'est donc dans cette optique que nous abordons la relaxation forte du problème. La contribution théorique de ce travail provient de la dérivation de deux nouvelles formulations mathématiques du problème "relaxé". La première ne contient que des variables de flot, tandis que l'autre est une restriction de la première à laquelle nous ajoutons des contraintes de capacité. Nous définissons ainsi un sous-problème sur un domaine réalisable restreint du problème initial. La structure de réseau, propagée dans ces deux formulations, nous permet alors de développer une méthode de restriction, à laquelle se greffe une adaptation d'un algorithme de relaxation primal-dual. L'intérêt de l'approche proposée découle directement des formulations dérivées qui nous permettent de traiter en un seul problème les problèmes de localisation et de flot. Il s'agit d'une heuristique qui génère une borne supérieure sur la valeur optimale de la fonction objectif du problème initial.

Nous avons développé deux versions algorithmiques : globale et locale, chacune d'elle déterminée par le moment et la façon de modifier les domaines réalisables

restreints. Dans la première, nous effectuons un changement global de ces derniers, impliquant un ajustement des coefficients d'ouverture des dépôts au fur et à mesure qu'il se produit une violation des contraintes de capacité. Dans l'approche locale, nous ne permettons qu'une modification des capacités pour les arcs dont la contrainte de capacité est violée, permettant ainsi une diminution du déficit total, contrairement à l'autre approche où il se crée des déficits. L'ajustement des coefficients d'ouverture se fera alors lorsqu'une solution primale-réalisable aura été trouvée, déterminant ainsi un nouveau sous-problème restreint à résoudre.

Les expérimentations ont été menées sur la version globale, celle que nous avons programmée, à partir de problèmes générés aléatoirement. Suite aux résultats numériques obtenus, exposés dans le dernier chapitre de ce mémoire, la méthode proposée semble prometteuse, quoique non-concluante. En effet, pour la majorité des problèmes testés, la solution fournie par l'algorithme, lorsque non optimale, constitue une borne supérieure située en-deça de 2,4% de la solution optimale du problème initial. Cependant, cette version algorithmique ne converge pas toujours. L'approche locale constitue alors une alternative de résolution qui devrait remédier à ce problème de non-convergence, puisqu'elle possède beaucoup moins de variabilité.

Les premiers tests numériques se révèlent encourageants et motivent la poursuite de nos travaux, qui consistent essentiellement à compléter et à améliorer les versions actuelles. La seconde version est déjà en voie d'implantation. Parmi les améliorations à apporter, outre certaines modifications qui permettront de rendre l'algorithme plus efficace, mentionnons en particulier les capacités sur le flot des arcs interdépôts ainsi que la solution de départ de l'algorithme. Nous savons pertinemment que l'algorithme de relaxation assure une meilleure convergence lorsque les bornes sur le flot des arcs sont serrées. Ainsi, il pourrait être bénéfique de resserrer davantage les bornes sur le flot de ces arcs de façon à tenir compte des coefficients d'ouverture des dépôts. Nous comptons également tester d'autres procédures d'initialisation afin de générer une meilleure solution de départ qui pourrait accélérer la convergence de l'algorithme.

Ce travail constitue donc un apport intéressant à l'étude du problème de localisation-distribution avec échanges interdépôts. En effet, nous avons dérivé deux

nouvelles formulations du modèle et développé une méthode de résolution originale. De plus amples travaux restent cependant à faire.

BIBLIOGRAPHIE

Aashtiani, H.A., and Magnanti, T.L. (1976). *Implementing Primal-Dual Network Flow Algorithm,* Massachussets Institute or Technology Operations Research Center Report 055-76.

Bertsekas, D.P. (1985). *A Unified Framework for Primal-Dual Methods in Minimum Cost Flow Problems,* **Mathematical Programming,** vol 32, no 2, 125-145.

Bertsekas, D., Hossein, P.A. and Tseng, P. (1987). *Relaxation Methods for Network Flow Problemes with Convex Arc Costs,* **Siam Journal Control and Optimization,** vol 25, 1219-1243.

Bertsekas, D.P., Hossein, P.A. and Tseng, P. (1985). *Relaxation Methods for Network Flow Problems with Convex Arc Costs,* rapport LIDS-P-1523, Massachussets Institute of Technology

Bertsekas, D.P. and Tseng, P. (1988). *Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problemes,* **Operations Research,** vol 36, no 1, 93-114.

Bertsekas, D.P. and Tseng, P. (1985). *RELAX : A Computer Code for Minimum Cost Network Flow Problems,* rapport LIDS Report-P-1469, Massachussets Institute of Technology.

Bertsekas, D.P. and Tseng, P. (1985). *Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problems,* rapport LIDS Report-P-1462, Massachussets Institute of Technology.

Bratley, P., Fox, B.L. and Schrage L.E., *A Guide to Simulation,* Springer-Verlag, New York, 1983.

Crainic, T.G., Dejax, P.J. et Delorme L. (1989). *Models for Multimode Multicommodity Location Problems with Interdepot Balancing Requirements,* **Annals of Operations Research** 18, 279-302.

Crainic, T.G, Dejax, P.J. et Delorme, L. (1986). *Problèmes de localisation-distribution avec échanges interdépôts : éléments de modélisation et propriétés,* publication #490, Centre de recherche sur les transports, Université de Montréal.

Dejax, P.J., Crainic, T.G. (1986). *A Review of Empty Flows and Fleet Management Models in Freight Transportation,* publication #482, Centre de recherche sur les transports, Université de Montréal.

Delorme, L., Crainic, T.G. et Dejax, P.J. (1988). *Dual Methods for Location/Distribution Problems with Balancing Requirements,* ORSA/TIMS Joint National Meeting, Washington, 26-27 avril.

De Tocqueville, E., *Modélisation d'un système de transport terrestre de conteneurs vides,* Rapport de stage de fin d'étude, Ecole Centrale des Arts et Manufactures, Paris, (1987).

Erlenkotter, D. (1978). *A Dual-Based Procedure for Uncapacitated Facility Location,* **Operations Research** 14 361-368.

Ferland, J.A., *Programmation non-linéaire,* Notes de cours, Département d'informatique et de recherche opérationnelle, Université de Montréal, 1980.

Ford, L.R. Jr. and Fulkerson, D.R, *Flows in networks,* Princeton University Press, Princeton, New Jersey, 1962.

Geoffrion, A.M., *Perspectives on Optimization : A Collection of Expository Articles,* A. M. Geoffrion, University of California, Los Angeles, 1972.

Geoffrion, A.M. (1971). *Duality in Nonlinear Programming : A Simplified Applications-Oriented Development,* **Siam Review,** vol 13 , no 1, 1-37.

Geoffrion, A.M. (1974). *Lagrangean Relaxation for Integer Programming,* **Mathematical Programming Study** 2, 82-114.

Grigoriadis, M.D. and Hsu, T. (1980). *The Rutgers Minimum Cost Network Flow Subroutines (RNET documentation).* Department of Computer Science, Rutgers University.

Hodgson, J. jr., *Application d'une méthode de "branch-and-bound" au problème de localisation avec transferts entre les dépôts,* Mémoire de maîtrise, Université de Montréal, Montréal, 1988.

Khumawala, B.M. (1972). *An Efficient Branch and Bound Algorithm for the Warehouse Location Problem,* **Management Science** 18B, 718-731.

Lafond, L., *Programmation non-linéaire,* Notes de cours, Département d'informatique et de recherche opérationnelle, Université de Montréal, 1985.

Lawler, E.L., *Combinatorial Optimization : Networks and Matroids,* Holt, Rinehart & Winston, New York, 1976.

Luenberger, D.G., *Introduction to Linear and Nonlinear Programming,* Addison-Wesley, 1973.

Rockafellar, R.T., *Network Flows and Monotropic Optimization,* J. Wiley-Interscience, New York, 1984.

Rockafellar, R.T., *Convex analysis,* Princeton University-Press, 1970.

Schrage, L. (1978). *Implicit Representation of Generalized Variable Upper Bounds in Linear Programming,* **Mathematical Programming Study** 14, 11-20.

Schrage, L. (1975). *Implicit Representation of Variable Upper Bound in Linear Programming,* **Mathematical Programming Study** 4, 119-132.

Shapiro, J.F. (1977). *A Note on the Primal-Dual and Out-of-kilter Algorithms for Network Optimization Problems,* **Networks** 7, 81-88.

Tétreault, N., Delorme, L. et Crainic, T.G. (1989). *Primal-Dual Methods for Location Problems with Balancing Requirements,* CORS/TIMS/ORSA Joint National Meeting, Vancouver, 8-10 mai.

Tseng, P., *Relaxation Methods for Monotropic Programming Problems,* Ph.D, Thesis, Operations Research Center, M.I.T, 1986.

Tseng, P. and Bertsekas, D.P. (1987). *Relaxation Methods for Linear Programs,* **Mathematics of Operations Research,** vol 12, no 4, 569-596.

APPENDICES

Appendice A	Rappels mathématiques	xiv
Appendice B	Itérations des algorithmes de relaxation primaux-duaux pour un déficit négatif	xix
Appendice C	Itération de l'algorithme de restriction pour un déficit négatif	xxiv
Appendice D	Génération des problèmes-tests	xxviii

APPENDICE A

RAPPELS MATHEMATIQUES

Dans cet appendice, nous rappelons certaines notions mathématiques auxquelles nous avons recours dans ce mémoire. Elles proviennent principalement du livre de Luenberger (1973), de l'article de Geoffrion (1971) et des notes de cours de Ferland (1980) et Lafond (1985) sur la programmation non-linéaire.

Ensemble et fonction convexes

Un sous-ensemble $X \subset R^n$ est un *ensemble convexe* si pout tout couple de points $x_1, x_2 \in X$ et pour tout scalaire θ, $0 \leq \theta \leq 1$, le point $\theta x_1 + (1 - \theta) x_2 \in X$.

Une fonction f définie sur un ensemble convexe Γ est une *fonction convexe* si pour tout couple de points $x_1, x_2 \in \Gamma$ et tout scalaire θ, $0 \leq \theta \leq 1$, la relation suivante est vérifiée

$$f(\theta x_1 + (1 - \theta) x_2) \leq \theta f(x_1) + (1 - \theta) f(x_2).$$

Dérivées

Soit $f : X \rightarrow R$ une fonction convexe et $x \in X$, la *dérivée à droite* et la *dérivée à gauche* de f au point x, dénotés respectivement par $f'(x)^+$ et $f'(x)^-$, sont donnés par

$$f'(x)^+ = \lim_{\xi \downarrow 0} \frac{f(x + \xi) - f(x)}{\xi}, \quad f'(x)^- = \lim_{\xi \downarrow 0} \frac{f(x) - f(x - \xi)}{\xi}.$$

Soit $f : R^n \rightarrow R$ une fonction convexe et $x \in \text{dom } f$, la *dérivée directionnelle* de f au point x dans la direction d, dénotée par $f'(x; d)$, est définie par

$$f'(x; d) = \lim_{\delta \rightarrow 0} \frac{f(x + \delta d) - f(x)}{\delta}.$$

Fonction séparable

Une fonction $f(x) : R^n \rightarrow R$ est une *fonction séparable* si elle peut s'écrire comme la somme de fonctions, chacune d'une seule composante du vecteur x, c'est-à-dire si on a

$$f(x) = \sum_{i=1}^{n} f_i(x_i).$$

Dualité non-linéaire

Soit le problème (primal) de programmation mathématique suivant :

(P) *Minimiser* $f(x)$

sujet à

$$g_i(x) \leq 0, \quad i = 1, 2, \cdots, m$$

$$x \in X$$

où chaque f et chaque $g_i(x)$ sont des fonctions réelles définies sur $X \subseteq R^n$. X est un sous-ensemble convexe non-vide sur lequel toutes les fonctions sont convexes.

Définition A.1 :

La *valeur optimale* du problème primal est le *infimum* de la fonction $f(x)$ sujet à $x \in X$ et $g_i(x) \leq 0, \forall i = 1, 2, ..., m$.

De même, la *valeur optimale* du problème dual est le *supremum* de $g(\lambda)$ sujet à $\lambda_i \geq 0, \forall i = 1, 2, ..., m$.

Définition A.2 :

Un couple (x^0, λ^0) satisfait les conditions d'optimalité pour (P) si

$$i) \quad x^0 \text{ minimise } f(x) + \sum_{i=1}^{m} \lambda_i \, g_i(x) \text{ sur } X$$

$$ii) \quad \sum_{i=1}^{m} \lambda_i \, g_i(x^0) \leq 0 \qquad \forall \, i = 1, 2, \ldots, m$$

$$iii) \quad \lambda_i \geq 0 \qquad \forall \, i = 1, 2, \ldots, m$$

λ^0 est dit un *vecteur de multiplicateurs optimaux* pour (P) s'il existe un x^0 tel que le couple (x^0, λ^0) satisfait les conditions d'optimalité pour (P).

Définition A.3 :

La *fonction de perturbation* v associée au problème (P) est la fonction dont le domaine est R^m et qui se définit comme suit

$$v(y) = \underset{x \in X}{infimum} \left\{ f(x) : g_i(x) \leq y_i, \; i = 1, 2, \cdots, m \right\}$$

où y est appelé le *vecteur de perturbation*.

Définition A.4 :

Soit ϕ une fonction convexe sur un ensemble $X \in R^n$ et $\bar{x} \in X$ un point où ϕ prend une valeur finie. Un vecteur $\gamma \in R^n$ est un *sous-gradient* de ϕ au point $\bar{x}$ si pour tout $x \in X$

$$\phi(x) \geq \phi(\bar{x}) + \gamma^T \, (x - \bar{x})$$

où γ^T représente le vecteur transposé de γ.

Définition A.5 :

Le problème est *stable* si $v(0)$ prend une valeur finie et s'il existe un scalaire M tel que pour tout $y \neq 0$

$$\frac{v(0) - v(y)}{||y||} \leq M.$$

Le problème dual lagrangien associé au problème (P) s'écrit

$$(D) \qquad \underset{\lambda \geq 0}{Maximiser} \left\{ \underset{x \in X}{infimum} \left\{ f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \right\} \right\}.$$

Théorème (de dualité forte) :

Si le problème (P) est stable, $X \subset R^n$ est un ensemble convexe et $g_1, g_2, \cdots, g_m$ sont des fonctions convexes sur X, alors

a) (D) possède une solution optimale;

b) la valeur optimale de (P) est égale à la valeur optimale de (D);

c) λ^* est une solution optimale de (D) si et seulement si $-\lambda^*$ est un sous-gradient de v à $y = 0$;

d) si λ^* est une solution optimale de (D) alors x est une solution optimale de (P) si et seulement si (x, λ^*) satisfait les conditions d'optimalité a), b), et c).

APPENDICE B

ITERATIONS DES ALGORITHMES DE RELAXATION PRIMAUX-DUAUX POUR UN DEFICIT NEGATIF

Dans cet appendice, nous détaillons une itération de l'algorithme de relaxation primal-dual pour les coûts linéaires (Bertsekas et Tseng, 1988) de même qu'une itération de l'algorithme pour les coûts convexes (Bertsekas, Hossein et Tseng, 1987) toutes deux construites à partir d'un noeud ayant un déficit négatif.

Itération de l'algorithme de relaxation (coûts linéaires)

Etape 0. Soit une paire primale-duale de vecteurs (x,π) satisfaisant les (CC').

Etape 1. Si $d_i = 0 \ \forall i \in N$ alors *STOP*.
Sinon faire : $M = \emptyset$
$S = \emptyset$
où
M = ensemble des sommets marqués non-examinés
S = ensemble des sommets marqués et examinés.
Choisir un sommet i tel que $d_i < 0$.
Faire : $M = \{i\}$
Aller à l'étape 2.

Etape 2. Choisir un sommet $k \in M$.
Faire : $S \leftarrow S \cup \{k\}$
$M \leftarrow M - \{k\}$.

Etape 3. Examiner le sommet k en marquant tous les sommets non-marqués $j \in \Gamma(k)$ et en leur donnant l'étiquette k de la manière suivante :

$$M \leftarrow M \cup \{j\} \quad \text{si} \quad \begin{cases} (k,j) \in A \mid x_{kj} < u_{kj} \text{ et } (k,j) \text{ libre} \\ \text{ou} \\ (j,k) \in A \mid x_{jk} > l_{jk} \text{ et } (j,k) \text{ libre.} \end{cases}$$

Si $C(S,\pi) > 0$ aller à l'étape 5.

Sinon

s'il existe un sommet $j \in M$ étiqueté à partir de k *t.q.* $d_j > 0$ aller à l'étape 4.

Sinon retourner à l'étape 2.

Etape 4. Retracer la (CAF) reliant les sommets j et i (j est le sommet marqué avec $d_j > 0$). Soit δ la capacité de la chaîne. Diminuer de δ le flot sur les arcs orientés de j vers i et ajouter δ unités de flot sur les arcs orientés de i vers j. Ajuster les déficits aux extrémités de la chaîne. Passer à l'itération suivante.

Etape 5. Soit $\bar{S} = N - S$, ($\bar{S} \neq \emptyset$ car $d_i < 0$ et $d_j \leq 0 \; \forall \, j \in S$ et $\sum_{i \in N} d_i = 0$)

Faire : $x_{kj} \leftarrow u_{kj} \quad \forall \, (k,j) \in (S,\bar{S})$ libre et $k \in M$

$x_{jk} \leftarrow l_{jk} \quad \forall \, (j,k) \in (S,\bar{S})$ libre et $k \in M$.

Faire $p_k \leftarrow p_k + \theta \quad \forall \, k \in S$ où θ est donné par (2.5).

Passer à l'itération suivante.

Itération de l'algorithme de relaxation (coûts convexes)

Etape 0. Soient x et π satisfaisant (ε–CC).

Etape 1. Si $d_i = 0 \ \forall i \in N$ alors *STOP*.
Sinon faire : $M = \varnothing$
$S = \varnothing$

où

M = ensemble des sommets marqués non-examinés

S = ensemble des sommets marqués et examinés.

Choisir un sommet i tel que $d_i < 0$.

Aller à l'étape 2.

Etape 2. Choisir un sommet $k \in M$.
Faire : $S \leftarrow S \cup \{k\}$
$M \leftarrow M - \{k\}$.
Aller à l'étape 3.

Etape 3. Examiner le sommet k en marquant tous les sommets non-marqués $j \in \Gamma(k)$ et en leur donnant l'étiquette k de la manière suivante :

$$M \leftarrow M \cup \{j\} \quad \text{si} \quad \begin{cases} (k, j) \in A \mid x_{kj} < u^{\varepsilon}_{kj} \\ \text{ou} \\ (j, k) \in A \mid x_{jk} > l^{\varepsilon}_{jk} \end{cases}$$

Si $C_{\varepsilon}(S, \pi) > 0$ aller à l'étape 5.

Sinon

s'il existe un sommet $j \in M$ étiqueté à partir de k *t.q.* $d_j > 0$ aller à l'étape 4.

Sinon retourner à l'étape 2.

Etape 4. Retracer la ε–*CAF* reliant les sommets j et i (j est le sommet marqué avec $d_j > 0$). Soit δ la capacité de la chaîne. Diminuer de δ le flot sur les arcs orientés de j vers i et ajouter δ unités de flot sur les arcs orientés de i vers j. Ajuster les déficits aux extrémités de la chaîne. Passer à l'itération suivante.

Etape 5. Déteminer λ^* *t.q.*

$$q(\pi - \lambda^* \mu(S)) = \min_{\lambda > 0} \{q(\pi - \lambda \mu(S)\}.$$

où

$$\mu_i(S) = \begin{cases} -1 & \text{si } i \in S \\ 0 & \text{si } i \in \overline{S} \end{cases}$$

Faire $\pi \leftarrow \pi - \lambda^* \mu(S)$. Mettre à jour des bornes–ε ainsi que le vecteur de flot afin de satisfaire (ε–*CC*).
Passer à l'itération suivante.

APPENDICE D

GENERATION DES PROBLEMES-TEST

Dans cet appendice, nous décrivons de façon algorithmique comment s'est faite la construction des problèmes tests. Nous avons écrit en *FORTRAN* le programme pour générer les données qui servent à tester notre algorithme (voir Hodgson, 1988 pour une description sur le format des données). Les différents paramètres du réseau sont générés selon une loi *Uniforme*(x, y), que nous dénoterons $U(x, y)$ dans ce qui suit. Dans notre programme, la procédure de cette loi statistique provient de :

A Guide for Simulation
Bratley, Fox et Schrage, 1983.

Le réseau de base

Nous considérons comme réseau de base, l'Amérique du Nord comportant 263 noeuds associés à différentes villes du Canada, des Etats-Unis et du Mexique. Posons Λ : le nombre total de noeuds de ce réseau.

La taille des problèmes

Soit

N : le nombre de noeuds du sous-réseau à générer
n : le nombre de clients
m : le nombre de dépots candidats
l : le nombre de produits
r : le nombre de zones

où

$N = 50 \Rightarrow$ petit problème
$N = 150 \Rightarrow$ moyen problème
$N = 250 \Rightarrow$ gros problème .

Pour chacun des sous-réseaux, environ un noeud sur six identifie un dépôt, tandis que tous les autres noeuds correspondent à des clients. La relation suivante : $n + m = N$ lie les defférents noeuds du sous-réseau entre eux. Le nombre de

produits peut varier entre un et dix pour chacun des problèmes générés. Quant au facteur zone, il permet de simuler le déséquilibre dans le réseau. Tous ces paramètres doivent être fournis par l'usager.

Les noeuds du sous-réseau

La construction du sous-réseau se fait en générant N noeuds parmi Λ selon une loi $U(1, N+1)$. De plus, pour chacune des zones z, nous générons m_z dépôts candidats selon une loi $U(1, N_z+1)$

où

N_z : le nombre de noeuds dans la zone z, $z = 1, \cdots, r$ avec $\sum_{z=1}^{p} N_z = N$

m_z : le nombre de dépôts candidats dans la zone z

n_z : le nombre de clients dans la zone z où $n_z = N_z - m_z$

L'offre et la demande totales par zone et par produit

Pour chacun des produits p, $p = 1, \cdots, l$ et chacune des zones z, $z = 1, \cdots, r$:

Nous générons l'offre totale et la demande totale selon des lois $U(0, \Delta_z^p)$ et $U(0, \Delta_z^p)$ respectivement

où

O_z^p : l'offre totale pour le produit p dans la zone z

D_z^p : la demande totale pour le produit p dans la zone z

Δ_z^p : une borne supérieure sur l'offre et la demande du produit p dans la zone z

Si $\dfrac{| O_z^p - D_z^p |}{O_z^p} < \varepsilon$, alors nous générons de nouvelles valeurs.

De plus, nous avons que :

$$\sum_{p=1}^{l} \sum_{z=1}^{r} D_z^p = \sum_{p=1}^{l} \sum_{z=1}^{r} O_z^p.$$

et nous supposons que :

$$\Delta_z^p = n_z * 5.$$

L'offre et la demande par client et par produit

Soit

O_i^p : l'offre du client i pour le produit p

D_i^p : la demande du client i pour le produit p

Nous faisons les hypothèses suivantes :

1) 80% de l'offre totale d'une zone est contrôlée par 20% des clients de cette zone.

2) idem pour la demande, mais pas nécessairement pour les mêmes clients.

Ainsi, pour pour chacune des zones :

- il existe deux classes de clients, soit $k = 1, 2$.
- Nous choisissons $0.2n_z$ clients pour la classe 1 selon une loi $U(1, n_z+1)$. Les autres clients appartiennent à la classe 2.
- Nous générons l'offre des clients de classe 1 selon une loi $U\left[0, \frac{0.8\ O_z^p}{0.2n_z}\right]$ et l'offre des clients de classe 2 selon une loi $U\left[0, \frac{0.2\ O_z^p}{0.8n_z}\right]$ pour chacun des produits p.

De plus, la somme des offres de chacun des clients dans une zone doit être égale à la somme totale des offres de la zone.

Nous générons la demande de la même façon et divisons à nouveau les clients en deux classes.

Les coûts de transport

Soit

c_{ijp} : coût unitaire de transport du client i au dépôt j pour le produit p
c_{jip} : coût unitaire de transport du dépôt j au client i pour le produit p
s_{kjp} : coût unitaire de transport du dépôt k au dépôt j pour le produit p.

Nous supposons que :

$$c_\alpha = \begin{cases} \text{distance du client } i \text{ au dépôt } j & \text{si } \alpha = ijp \\ \text{distance du dépôt } j \text{ au client } i & \text{si } \alpha = jip \\ \lambda * (\text{distance entre les dépôts } k \text{ et } j) & \text{si } \alpha = kjp \end{cases}$$

et faisons l'hypothèse suivante :

Si la distance entre les dépôts k et j est supérieure à 3 fois la distance entre le client i et le dépôt j alors les mouvements du client i au dépôt j et du dépôt j au dépôt k seront plus économiques que les mouvements directs entre le client i et le dépôt k. Alors $\lambda = 0{,}72$.

Coûts fixes

Soit

f_j : le coût d'ouverture du dépôt j
$\bar{f}$: le coût moyen d'ouverture d'un dépôt (à suppléer par l'usager)

Nous générons le coût fixe pour chacun des dépôts selon une loi $U\left[\frac{\bar{f}}{2}, \frac{\bar{f}}{3}\right]$.

REMERCIEMENTS

Je tiens à remercier mon directeur de maîtrise, M. Teodor Crainic, professeur agrégé à l'Université du Québec et chercheur au Centre de recherche sur les transports (C.R.T.), pour m'avoir proposé ce sujet de recherche très intéressant, pour ses conseils et ses critiques constructives ainsi que pour son appui financier. L'assistance et la confiance qu'il m'a témoignées tout au long de ce travail constituent pour moi une expérience enrichissante.

Je remercie également M. Louis Delorme, maintenant chercheur à l'I.R.E.Q., pour l'aide précieuse qu'il m'a apportée ainsi que pour son support.

Je voudrais aussi remercier le Gouvernement français, qui dans le cadre d'un projet intégré franco-québécois, m'a donné une bourse pour effectuer un stage de trois mois en France. Plus particulièrement, je remercie M. Pierre Dejax, professeur de génie industriel et directeur du Laboratoire Economique, Industriel et Social de l'Ecole Centrale de Paris, pour l'encadrement et l'accueil chaleureux qui furent très appréciés. Merci aussi à sa femme, Sylvianne.

Je tiens à remercier le C.R.T. pour les bourses qu'il m'a accordées tout au long de mes études de maîtrise.

Finalement, j'aimerais remercier ma famille ainsi que Jean-François pour leur soutien constant, leur patience et leurs encouragements du début à la fin. C'est donc, pour moi et pour eux, avec grand soulagement que je dépose ce mémoire.